Setup Superset

1. Task requirement

To run the superset with a podman play kube which is generated with the helm chart of the superset.

Superset is a modern data exploration and data visualisation platform. Superset can replace or augment proprietary business intelligence tools for many teams. Superset integrates well with a variety of data sources. Superset provides: A no-code interface for building charts quickly.

2. Environment details

- OS: Ubuntu 20.04

My System configuration

- CPU: Intel Core i5-8350U CPU @ 1.70GHz x 8
- RAM: 8GB (4GB x 2 SODIMM DDR4)
- Storage: 512GB

3. List of tools and technologies

- Podman version 3.4.2
- Helm version 3.12.1

Others

- Vim
- Bash

4. Definition of tools

- **Podman** - It is an open source tool for developing, managing, and running containers on your Linux systems.
- **Helm** - It is a tool that streamlines installing and managing Kubernetes applications.
- Kubernetes -
- **Vim** - It is a highly configurable text editor built to make creating and changing any kind of text very efficient
- **Bash** - It is the command line shell that you encounter when you open the terminal on most Unix operating systems, like MacOS and Linux.

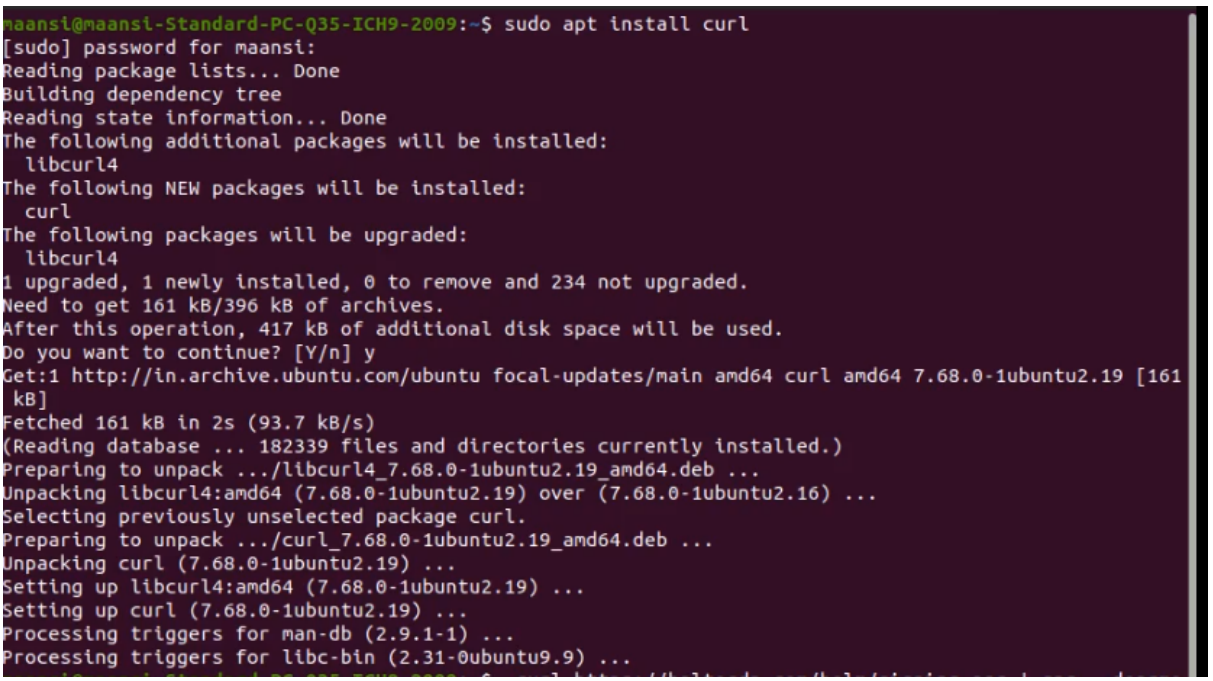
5. Command for the setup or configuration

Step 1. Run the following command to install curl.

```
sudo apt install curl
```

- curl is a command-line tool that allows you to fetch data from the internet. It's like a web browser for your terminal. You can use it to download files, make web requests, and interact with web services directly from the command line.

Output:



Step 2. Adding Helm GPG Key.

```
$ curl https://baltocdn.com/helm/signing.asc | gpg --dearmor | sudo tee /usr/share/keyrings/helm.gpg > /dev/null
```

- **`curl <https://baltocdn.com/helm/signing.asc>:`**
  curl : To download the contents of the specified URL, which appears to be the GPG (GNU Privacy Guard) signing key for Helm, a package manager for Kubernetes.

- **`gpg --dearmor`:** In GPG, "dearmor" means to convert a GPG public or private key from the binary format to a text-based format. This is often done for distribution or storage purposes.

- **`sudo tee /usr/share/keyrings/u2`helm.gpg > /dev/null` :** This part of the command uses tee to write the output of the previous gpg command to the file **`/usr/share/keyrings/helm.gpg`**.

- **`> /dev/null :`** redirects the standard output (stdout) to the "null" device .

Output:

```
maansi@indianrenters-Latitude-5490:~$ curl https://baltocdn.com/helm/signing.asc | gpg --dearmor | sudo tee /usr/share/keyrings/helm.gpg > /dev/null
[sudo] password for maansi:   % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  1699  100  1699    0     0   5838      0 --:--:-- --:--:-- --:--:--  5838

maansi@indianrenters-Latitude-5490:~$
```

Step 3. Adding Helm Repository to Package Sources.

```
$ echo "deb [arch=$(dpkg --print-architecture)
sudsigned-by=/usr/share/keyrings/helm.gpg]
https://baltocdn.com/helm/stable/debian/ all main" | sudo tee
/etc/apt/sources.list.d/helm-stable-debian.list
```

- echo : display the text/string on the terminal
- line tells the apt package manager to download Helm packages from the https://baltocdn.com/helm/stable/debian/ repository.
- The arch=$(dpkg --print-architecture) : apt package manager to download the correct package for our system architecture.
- The signed-by=/usr/share/keyrings/helm.gpg part tells the apt package manager to verify the authenticity of the packages using the GPG key that is stored in the file /usr/share/keyrings/helm.gpg.
- Creates a new file called helm-stable-debian.list in the directory /etc/apt/sources.list.d/.
- It adds the following line to the file: deb [arch=(dpkg --print-architecture) signed-by=/usr/share/keyrings/helm.gpg] https://baltocdn.com/helm/stable/debian/ all main

Output:

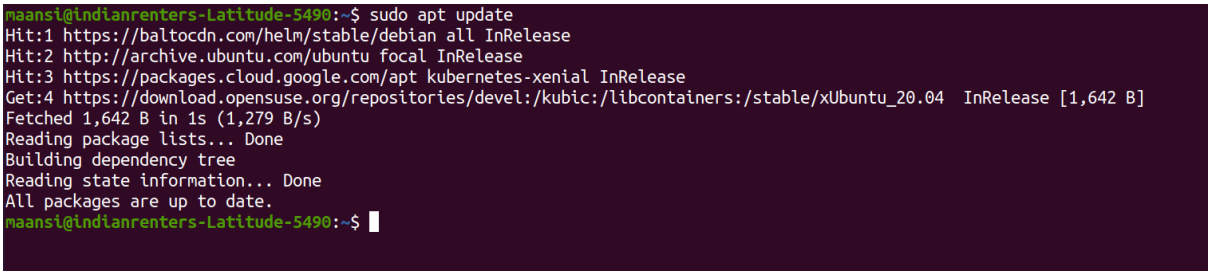
```
maansi@indianrenters-Latitude-5490:~$ echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/helm.gpg] https://baltocdn.com/helm/sta
ble/debian/ all main" | sudo tee /etc/apt/sources.list.d/helm-stable-debian.list
deb [arch=amd64 signed-by=/usr/share/keyrings/helm.gpg] https://baltocdn.com/helm/stable/debian/ all main
maansi@indianrenters-Latitude-5490:~$
```

Step 4. Updating Packages and Repositories.

```
$ sudo apt update
```

- **sudo**: This stands for "Superuser Do" and is used to execute commands with superuser (administrator) privileges. It allows you to perform actions that require elevated permissions.
- **apt**: This stands for "Advanced Package Tool." It's a command-line tool used to manage software packages on Debian-based systems. You can use it to install, update, remove, and manage software packages.
- **update**: This is a subcommand of apt that instructs the package manager to update the package information. The package information is a database of available software packages, their versions, and where to download them from. This database needs to be refreshed periodically to ensure that your system has the latest information about available software.

Output:



Step 5. Install helm.

```
$ sudo apt install helm
```

Or

```
$ sudo snap install helm --classic
```

- sudo: This prefix is used to execute the command with superuser (administrator) privileges. It allows you to install software system-wide.
- apt: The package management tool used for managing software packages on Debian-based systems.

- install: A command to instruct apt to install a specified package.
- helm: The name of the package you want to install. In this case, "Helm" is a tool used for managing Kubernetes applications. Helm simplifies the process of deploying and managing complex applications on Kubernetes clusters.

Output:

```
mansi@mansi-Standard-PC-Q35-ICH9-2009:~$ sudo apt install helm
Reading package lists... 0%
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  helm
0 upgraded, 1 newly installed, 0 to remove and 218 not upgraded.
Need to get 16.0 MB of archives.
After this operation, 50.6 MB of additional disk space will be used.
Get:1 https://baltocdn.com/helm/stable/debian all/main amd64 helm amd64 3.12.3-1 [16.0 MB]
Fetched 16.0 MB in 3s (4,813 kB/s)
Selecting previously unselected package helm.
(Reading database ... 179517 files and directories currently installed.)
Preparing to unpack .../helm_3.12.3-1_amd64.deb ...
Unpacking helm (3.12.3-1) ...
Setting up helm (3.12.3-1) ...
Processing triggers for man-db (2.9.1-1) ...
```

Step 6. Verifying the installation and version.

```
$ helm version
```

Output:

```
mansi@mansi-Standard-PC-Q35-ICH9-2009:~$ helm version
version.BuildInfo{Version:"v3.12.3", GitCommit:"3a31588ad33fe3b89af5a2a54ee1d25bfe6eaa5e", GitTreeState:"clean", GoVersion:"go1.20.7
"}
mansi@mansi-Standard-PC-Q35-ICH9-2009:~$
```

Step 7. Create directory

```
# mkdir superset-poc
```

- mkdir: It's used to create a new directory (folder) in the file system.
- superset-poc: This is the name of the directory you want to create.

Output:

Step 8. Go in to directory

```
# cd superset-poc
```

- cd: This is a command that stands for "change directory."

- superset-poc: This is the name of the directory you want to navigate to.

Output:

```
mansi@mansi-Standard-PC-Q35-ICH9-2009:~$ cd superset-poc
mansi@mansi-Standard-PC-Q35-ICH9-2009:~/superset-poc$
```

Step 9. Run the command to install vim.

```
$ sudo apt install vim
```

Vim is a text editor mainly used for writing and editing text and code. It's efficient, customizable, and works in terminal environments. It's great for programmers, system administrators, and anyone who wants fast and powerful text editing.

Output:

```
maansi@maansi-Standard-PC-Q35-ICH9-2009:~/superset-poc$ sudo apt install vim
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  vim-common vim-runtime vim-tiny
Suggested packages:
  ctags vim-doc vim-scripts indent
The following NEW packages will be installed:
  vim vim-runtime
The following packages will be upgraded:
  vim-common vim-tiny
2 upgraded, 2 newly installed, 0 to remove and 232 not upgraded.
Need to get 7,116 kB/7,785 kB of archives.
After this operation, 34.6 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 vim-runtime all 2:8.1.2269-1ubuntu
5.18 [5,875 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 vim amd64 2:8.1.2269-1ubuntu5.18 [
1,242 kB]
Fetched 7,116 kB in 3s (2,522 kB/s)
(Reading database ... 182346 files and directories currently installed.)
Preparing to unpack .../vim-tiny_2%3a8.1.2269-1ubuntu5.18_amd64.deb ...
Unpacking vim-tiny (2:8.1.2269-1ubuntu5.18) over (2:8.1.2269-1ubuntu5.11) ...
Preparing to unpack .../vim-common_2%3a8.1.2269-1ubuntu5.18_all.deb ...
Unpacking vim-common (2:8.1.2269-1ubuntu5.18) over (2:8.1.2269-1ubuntu5.11) ...
Selecting previously unselected package vim-runtime.
Preparing to unpack .../vim-runtime_2%3a8.1.2269-1ubuntu5.18_all.deb ...
Adding 'diversion of /usr/share/vim/vim81/doc/help.txt to /usr/share/vim/vim81/doc/help.txt.vim-tiny
by vim-runtime'
Adding 'diversion of /usr/share/vim/vim81/doc/tags to /usr/share/vim/vim81/doc/tags.vim-tiny by vim-r
untime'
Unpacking vim-runtime (2:8.1.2269-1ubuntu5.18) ...
Selecting previously unselected package vim.
Preparing to unpack .../vim_2%3a8.1.2269-1ubuntu5.18_amd64.deb ...
Unpacking vim (2:8.1.2269-1ubuntu5.18) ...
Setting up vim-common (2:8.1.2269-1ubuntu5.18) ...
Setting up vim-runtime (2:8.1.2269-1ubuntu5.18) ...
Setting up vim (2:8.1.2269-1ubuntu5.18) ...
update-alternatives: using /usr/bin/vim.basic to provide /usr/bin/vim (vim) in auto mode
update-alternatives: using /usr/bin/vim.basic to provide /usr/bin/vimdiff (vimdiff) in auto mode
update-alternatives: using /usr/bin/vim.basic to provide /usr/bin/rvim (rvim) in auto mode
update-alternatives: using /usr/bin/vim.basic to provide /usr/bin/rview (rview) in auto mode
update-alternatives: using /usr/bin/vim.basic to provide /usr/bin/vi (vi) in auto mode
update-alternatives: using /usr/bin/vim.basic to provide /usr/bin/view (view) in auto mode
update-alternatives: using /usr/bin/vim.basic to provide /usr/bin/ex (ex) in auto mode
Setting up vim-tiny (2:8.1.2269-1ubuntu5.18) ...
Processing triggers for desktop-file-utils (0.24-1ubuntu3) ...
Processing triggers for mime-support (3.64ubuntu1) ...
Processing triggers for hicolor-icon-theme (0.17-2) ...
Processing triggers for gnome-menus (3.36.0-1ubuntu1) ...
Processing triggers for man-db (2.9.1-1) ...
maansi@maansi-Standard-PC-Q35-ICH9-2009:~/superset-poc$
```

Step 10. Create a yaml file and add details in it .

First I check hosts file and add the domain.

```
sudo vim /etc/hosts
```

And add :
192.168.1.254 fosteringlinux.com and save the file.

Create values-stage.yaml file

```
vim values-stage.yaml
```

- vim: This is a command-line text editor that allows you to create, view, and edit text files directly within the terminal.
- value-stage.yaml: This is the name of the file you will open.

```
global:
  hosts:
    domain: fosteringlinux.com
postgresql:
  image:
    tag: 14.0
certmanager-issuer:
  email: mansi_01@fosteringlinux.com
```

Note :
"In the values-stage.yaml file, modify the domain and email according to your specific requirements.To save a file in Vim, press Esc and then type :wq! followed by Enter"

"When you copy and paste to another place, the indentation gets messed up, so pay attention to the indentation."

Step 11. Creating Superset Deployment Configuration

```
#  helm template --dry-run --debug superset/superset --generate-name
--values values-stage.yaml > superset-kube.yaml
```

- helm template: This is the command that tells Helm to generate Kubernetes manifests based on a Helm chart.

- --dry-run: This indicates Helm will simulate the installation without actually deploying anything. This is useful for seeing what manifests would be generated without affecting the cluster.

- --debug: This enables debug output, providing more detailed information about the Helm template process.

- superset/superset: This specifies the Helm chart to use for generating the Kubernetes manifests. The chart name is in the format repository/chart-name.

- --generate-name: This generates a unique name for the release based on the chart's name. It's used when you want to create a new release without specifying a release name.

- --values values-stage.yaml: This points to a values file (values-stage.yaml) that provides custom configuration values for the Helm chart.

- > superset-kube.yaml: This part of the command uses the output redirection (>) to save the generated Kubernetes manifests to a file named superset-kube.yaml.

  In summary, this command generates Kubernetes manifests for deploying the Superset application using a specific Helm chart and custom configuration values from the `values-stage.yaml` file. The generated manifests are saved in the `superset-kube.yaml` file. The use of `--dry-run` and `--debug` ensures that we can preview the manifests before actually deploying the application, it helps to verify that the configuration is correct.

Output :

```
mansi@mansi-Standard-PC-Q35-ICH9-2009:~/superset-poc$  helm template --dry-run --debug superset/superset --generate-name --values va
lues-stage.yaml > superset-kube.yaml
install.go:200: [debug] Original chart version: ""
Error: repo superset not found
helm.go:84: [debug] repo superset not found
helm.sh/helm/v3/pkg/downloader.pickChartRepositoryConfigByName
        helm.sh/helm/v3/pkg/downloader/chart_downloader.go:350
helm.sh/helm/v3/pkg/downloader.(*ChartDownloader).ResolveChartVersion
        helm.sh/helm/v3/pkg/downloader/chart_downloader.go:253
helm.sh/helm/v3/pkg/downloader.(*ChartDownloader).DownloadTo
        helm.sh/helm/v3/pkg/downloader/chart_downloader.go:90
helm.sh/helm/v3/pkg/action.(*ChartPathOptions).LocateChart
        helm.sh/helm/v3/pkg/action/install.go:789
main.runInstall
        helm.sh/helm/v3/cmd/helm/install.go:212
main.newTemplateCmd.func2
        helm.sh/helm/v3/cmd/helm/template.go:88
github.com/spf13/cobra.(*Command).execute
        github.com/spf13/cobra@v1.6.1/command.go:916
github.com/spf13/cobra.(*Command).ExecuteC
        github.com/spf13/cobra@v1.6.1/command.go:1044
github.com/spf13/cobra.(*Command).Execute
        github.com/spf13/cobra@v1.6.1/command.go:968
main.main
        helm.sh/helm/v3/cmd/helm/helm.go:83
runtime.main
        runtime/proc.go:250
runtime.goexit
```

Problem 1. There I got the above error .

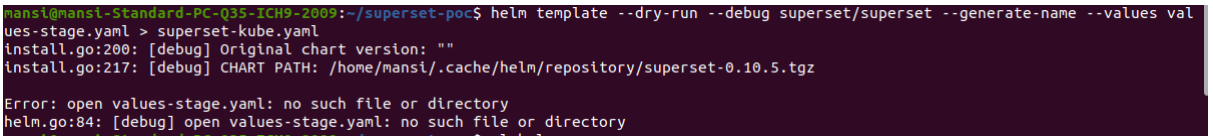
To solve this I run below commands.

```
$  helm repo add superset https://apache.github.io/superset
```

```
$  helm repo update
```

For this I follow this link:
https://stackoverflow.com/questions/66706363/where-is-the-superset-helm-chart

Problem 2.



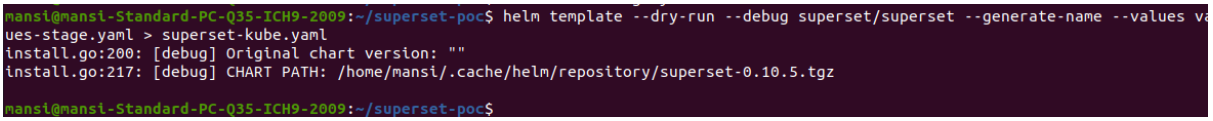
When I run this command I get the above error.

To solve this, I change the file name that is incorrect while creating the yaml file. The correct name is **values-stage.yaml** not **value-stage.yaml .**

Then I run the command.

```
# helm template --dry-run --debug superset/superset --generate-name
--values values-stage.yaml > superset-kube.yaml
```

Output:



Step 12. Edit superset-kube.yaml and create a new final.yaml as per the requirement in podman play kube.

```
cat superset-kube.yaml
```

```
vim final.yaml
```

Add the below data final.yaml file.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: kubepostgresql
spec:
```

```
replicas: 1
template:
    metadata:
    name: kubepostgresql
    spec:
    serviceAccountName: default
    securityContext:
    fsGroup: 1001
    hostNetwork: true
    hostIPC: false
    initContainers: []
    containers:
    - name: postgresql
          image: docker.io/bitnami/postgresql:14.6.0-debian-11-r13
          imagePullPolicy: "IfNotPresent"
          securityContext:
          runAsUser: 1001
          env:
          - name: BITNAMI_DEBUG
          value: "false"
          - name: POSTGRESQL_PORT_NUMBER
          value: "5432"
          - name: POSTGRESQL_VOLUME_DIR
          value: "/bitnami/postgresql"
          - name: PGDATA
          value: "/bitnami/postgresql/data"
          - name: POSTGRES_USER
          value: "superset"
          - name: POSTGRES_POSTGRES_PASSWORD
          value: "superset"
          - name: POSTGRES_PASSWORD
          value: "superset"
          - name: POSTGRES_DB
          value: "superset"
          - name: POSTGRESQL_ENABLE_LDAP
          value: "no"
          - name: POSTGRESQL_ENABLE_TLS
          value: "no"
          - name: POSTGRESQL_LOG_HOSTNAME
          value: "false"
          - name: POSTGRESQL_LOG_CONNECTIONS
          value: "false"
          - name: POSTGRESQL_LOG_DISCONNECTIONS
          value: "false"
          - name: POSTGRESQL_PGAUDIT_LOG_CATALOG
          value: "off"
```

```
            # Others
            - name: POSTGRESQL_CLIENT_MIN_MESSAGES
            value: "error"
            - name: POSTGRESQL_SHARED_PRELOAD_LIBRARIES
            value: "pgaudit"
            ports:
            - name: tcp-postgresql
            containerPort: 5432
            resources:
            limits: {}
            requests:
            cpu: 250m
            memory: 256Mi
            volumeMounts:
            - name: data
            mountPath: /bitnami/postgresql
      volumes:
      - name: data
            hostPath:
            path: /home/maansi/superset-poc/postgres/

---

apiVersion: apps/v1
kind: Deployment
metadata:
  name: kuberedis-master
spec:
  replicas: 1
  template:
      spec:
      securityContext:
      fsGroup: 1001
      terminationGracePeriodSeconds: 30
      containers:
      - name: redis
            image: docker.io/bitnami/redis:7.0.10-debian-11-r4
            imagePullPolicy: "IfNotPresent"
            securityContext:
            runAsUser: 1001
            command:
            - /bin/bash
            args:
            - -c
            - /opt/bitnami/scripts/start-scripts/start-master.sh
            env:
```

```
            - name: BITNAMI_DEBUG
            value: "false"
            - name: REDIS_REPLICATION_MODE
            value: "master"
            - name: ALLOW_EMPTY_PASSWORD
            value: "yes"
            - name: REDIS_TLS_ENABLED
            value: "no"
            - name: REDIS_PORT
            value: "6379"
            ports:
            - name: redis
            containerPort: 6379
            hostPort: 6379
            resources:
            limits: {}
            requests: {}
            volumeMounts:
            - name: start-scripts
            mountPath:
/opt/bitnami/scripts/start-scripts/start-master.sh
            subPath: start-master.sh
            - name: redis-data
            mountPath: /data
            - name: config
            mountPath: /opt/bitnami/redis/mounted-etc
      volumes:
      - name: start-scripts
            hostPath:
            path: /home/maansi/superset-poc/redis/start-master.sh
            type: FileOrCreate
            defaultMode: 0777
      - name: config
            hostPath:
            path: /home/maansi/superset-poc/redis/redis-conf
            type: Directory
      - name: redis-data
            hostPath:
            path: /home/maansi/superset-poc/redis/redis-data

---

apiVersion: apps/v1
kind: Deployment
metadata:
  name: kubesuperset-worker
```

```
spec:
  replicas: 1
  template:
      spec:
      securityContext:
      runAsUser: 0
      initContainers:
      - command:
            - /bin/sh
            - -c
            - dockerize -wait "tcp://192.168.122.113:5432" -wait
"tcp://192.168.122.113:6379" -timeout 120s
            env:
            - name: REDIS_HOST
            value: 192.168.122.113
            - name: REDIS_PORT
            value: "6379"
            - name: DB_HOST
            value: 192.168.122.113
            - name: DB_PORT
            value: "5432"
            - name: DB_USER
            value: "superset"
            - name: DB_PASS
            value: "superset"
            - name: DB_NAME
            value: "superset"
            image: 'apache/superset:dockerize'
            imagePullPolicy: 'IfNotPresent'
            name: wait-for-postgres-redis
      containers:
      - name: superset
            image:
"apachesuperset.docker.scarf.sh/apache/superset:2.1.0"
            imagePullPolicy: IfNotPresent
            command: ["/bin/sh","-c",".
/app/pythonpath/superset_bootstrap.sh; celery
--app=superset.tasks.celery_app:app worker"]
            env:
            - name: "SUPERSET_PORT"
            value: "8088"
            - name: REDIS_HOST
            value: 192.168.122.113
            - name: REDIS_PORT
            value: "6379"
            - name: DB_HOST
```

```
            value: 192.168.122.113
            - name: DB_PORT
            value: "5432"
            - name: DB_USER
            value: "superset"
            - name: DB_PASS
            value: "superset"
            - name: DB_NAME
            value: "superset"
            volumeMounts:
            - name: superset-config
            mountPath: "/app/pythonpath"
            readOnly: true
            livenessProbe:
            exec:
            command:
                  - sh
                  - -c
                  - celery -A superset.tasks.celery_app:app inspect ping
-d celery@$HOSTNAME
            failureThreshold: 3
            initialDelaySeconds: 120
            periodSeconds: 60
            successThreshold: 1
            timeoutSeconds: 60
            resources: {}
      volumes:
      - name: superset-config
            hostPath:
            path: /home/maansi/superset-poc/superset/

---

apiVersion: apps/v1
kind: Deployment
metadata:
  name: kubesuperset
spec:
  replicas: 1
  template:
      spec:
      securityContext:
      runAsUser: 0
      initContainers:
      - command:
            - /bin/sh
```

```
            - -c
            - dockerize -wait "tcp://192.168.122.113:5432" -timeout 120s
            env:
            - name: REDIS_HOST
            value: 192.168.122.113
            - name: REDIS_PORT
            value: "6379"
            - name: DB_HOST
            value: 192.168.122.113
            - name: DB_PORT
            value: "5432"
            - name: DB_USER
            value: "superset"
            - name: DB_PASS
            value: "superset"
            - name: DB_NAME
            value: "superset"
            image: 'apache/superset:dockerize'
            imagePullPolicy: 'IfNotPresent'
            name: wait-for-postgres
      containers:
      - name: superset
            image:
"apachesuperset.docker.scarf.sh/apache/superset:2.1.0"
            imagePullPolicy: IfNotPresent
            command: ["/bin/sh","-c",".
/app/pythonpath/superset_bootstrap.sh; /usr/bin/run-server.sh"]
            env:
            - name: "SUPERSET_PORT"
            value: "8088"
            - name: REDIS_HOST
            value: 192.168.122.113
            - name: REDIS_PORT
            value: "6379"
            - name: DB_HOST
            value: 192.168.122.113
            - name: DB_PORT
            value: "5432"
            - name: DB_USER
            value: "superset"
            - name: DB_PASS
            value: "superset"
            - name: DB_NAME
            value: "superset"
            volumeMounts:
            - name: superset-config
```

```
            mountPath: "/app/pythonpath"
            readOnly: true
            ports:
            - name: http
            containerPort: 8088
            hostPort: 8088
            protocol: TCP
            resources: {}
      volumes:
      - name: superset-config
            hostPath:
            path: /home/maansi/superset-poc/superset/
```

Note : There i replace the path .
Example:

```
path: /home/harsh/superset/redis/start-master.sh
 To this
path: /home/mansi/superset-poc/redis/start-master.sh
```

Also change the ip with your system ip.

"When you copy and paste to another place, the indentation gets messed up, so pay attention to the indentation."

Then save the file.

Step 13. Check list of file in current directory

```
# ls -lrth
```

The command ls -lrth is used to list the contents of a directory:

- -l: Detailed (long) format showing permissions, owner, size, and more.
- -r: Listing in reverse order.
- -t: Sorting by modification time (most recent first).
- -h: File sizes in a human-readable format.

Output:

```
mansi@mansi-Standard-PC-Q35-ICH9-2009:~/superset-poc$ ls -lrth
total 44K
-rw-rw-r-- 1 mansi mansi  120 Aug 17 13:34 values-stage.yaml
-rw-rw-r-- 1 mansi mansi  29K Aug 17 13:50 superset-kube.yaml
-rw-rw-r-- 1 mansi mansi 7.5K Aug 17 13:52 final.yaml
mansi@mansi-Standard-PC-Q35-ICH9-2009:~/superset-poc$
```

Step 14. Create a script file and add details in it .

```
vim pre-install-task.sh
```

1. Bash Script: A Bash script is a file containing a sequence of commands written in the Bash scripting language. These commands can include system commands, variable assignments, conditionals, loops, and more. Bash scripts are used to automate tasks, perform system administration, and execute a series of actions.

Add in the below script----------

```
#!/bin/bash

mkdir superset
mkdir redis
mkdir postgres

cat <<- END > superset/superset_bootstrap.sh
      #!/bin/bash
      if [ ! -f ~/bootstrap ]; then echo "Running Superset with uid 0" >
~/bootstrap; fi

END

cat <<- END > superset/superset_config.py
import os
from cachelib.redis import RedisCache

def env(key, default=None):
      return os.getenv(key, default)

MAPBOX_API_KEY = env('MAPBOX_API_KEY', '')
CACHE_CONFIG = {
      'CACHE_TYPE': 'RedisCache',
      'CACHE_DEFAULT_TIMEOUT': 300,
      'CACHE_KEY_PREFIX': 'superset_',
      'CACHE_REDIS_HOST': env('REDIS_HOST'),
      'CACHE_REDIS_PORT': env('REDIS_PORT'),
      'CACHE_REDIS_PASSWORD': env('REDIS_PASSWORD'),
      'CACHE_REDIS_DB': env('REDIS_DB', 1),
}
DATA_CACHE_CONFIG = CACHE_CONFIG

SQLALCHEMY_DATABASE_URI =
f"postgresql+psycopg2://{env('DB_USER')}:{env('DB_PASS')}@{env('DB_HOST'
)}:{env('DB_PORT')}/{env('DB_NAME')}"
SQLALCHEMY_TRACK_MODIFICATIONS = True
SECRET_KEY = env('SECRET_KEY', 'thisISaSECRET_1234')

class CeleryConfig(object):
      CELERY_IMPORTS = ('superset.sql_lab', )
      CELERY_ANNOTATIONS = {'tasks.add': {'rate_limit': '10/s'}}
      BROKER_URL = f"redis://{env('REDIS_HOST')}:{env('REDIS_PORT')}/0"
      CELERY_RESULT_BACKEND =
f"redis://{env('REDIS_HOST')}:{env('REDIS_PORT')}/0"
```

```
CELERY_CONFIG = CeleryConfig
RESULTS_BACKEND = RedisCache(
      host=env('REDIS_HOST'),
      port=env('REDIS_PORT'),
      key_prefix='superset_results'
)
END
cat <<- END > superset/superset_init.sh
#!/bin/sh
set -eu
echo "Upgrading DB schema..."
superset db upgrade
echo "Initializing roles..."
superset init

echo "Creating admin user..."
superset fab create-admin \
            --username admin \
                  --firstname Superset \
                  --lastname Admin \
                  --email admin@superset.com \
                  --password admin \
                  || true

if [ -f "/app/configs/import_datasources.yaml" ]; then
   echo "Importing database connections.... "
   superset import_datasources -p /app/configs/import_datasources.yaml
fi
END
mkdir redis/redis-conf
cat <<- END > redis/redis-conf/master.conf
      dir /data
      # User-supplied master configuration:
      rename-command FLUSHDB ""
      rename-command FLUSHALL ""
      # End of master configuration
END

cat <<- END > redis/redis-conf/redis.conf
      appendonly yes
      save ""
END

cat <<- END > redis/redis-conf/replica.conf
      dir /data
      rename-command FLUSHDB ""
```

```
      rename-command FLUSHALL ""
END

mkdir -p redis/redis-data
cat <<- END > redis/master-redis.sh
      #!/bin/bash
      [[ -f \$REDIS_PASSWORD_FILE ]] && export REDIS_PASSWORD="\$(<
"\${REDIS_PASSWORD_FILE}")"
      if [[ -f /opt/bitnami/redis/mounted-etc/master.conf ]];then
      cp /opt/bitnami/redis/mounted-etc/master.conf
/opt/bitnami/redis/etc/master.conf
      fi
      if [[ -f /opt/bitnami/redis/mounted-etc/redis.conf ]];then
      cp /opt/bitnami/redis/mounted-etc/redis.conf
/opt/bitnami/redis/etc/redis.conf
      fi
      ARGS=("--port" "\${REDIS_PORT}")
      ARGS+=("--protected-mode" "no")
      ARGS+=("--include" "/opt/bitnami/redis/etc/redis.conf")
      ARGS+=("--include" "/opt/bitnami/redis/etc/master.conf")
      exec redis-server "\${ARGS[@]}"
END
chmod +x superset/superset_init.sh
chmod +x superset/superset_config.py
chmod +x superset/superset_bootstrap.sh
```

And Save the file.

Step 15. Check list of file in current directory

```
# ls -lrth
```

Output:

```
mansi@mansi-Standard-PC-Q35-ICH9-2009:~/superset-poc$ ls -lrth
total 48K
-rw-rw-r-- 1 mansi mansi  120 Aug 17 13:34 values-stage.yaml
-rw-rw-r-- 1 mansi mansi  29K Aug 17 13:50 superset-kube.yaml
-rw-rw-r-- 1 mansi mansi 7.5K Aug 17 13:52 final.yaml
-rw-rw-r-- 1 mansi mansi 3.2K Aug 17 14:05 pre-install-task.sh
mansi@mansi-Standard-PC-Q35-ICH9-2009:~/superset-poc$
```

Step 16. Give permission

```
# chmod +x pre-install-task.sh
```

- **chmod :** change the permissions of a file or directory.
- **+x :** to add the execute permission to the file or directory.

Output:

```
mansi@mansi-Standard-PC-Q35-ICH9-2009:~/superset-poc$  chmod +x pre-install-task.sh
mansi@mansi-Standard-PC-Q35-ICH9-2009:~/superset-poc$
```

Step 17. Run the script

```
sh pre-install-task.sh
```

- sh: Command to run a shell interpreter.
- pre-install-task.sh: Name of the shell script file to be executed.

Step 18. Step 15. Check list of file in current directory.

```
# ls -lrth
```

Output :

```
maansi@maansi-Standard-PC-Q35-ICH9-2009:~/superset-poc$ ls -lrth
total 64K
-rw-rw-r-- 1 maansi maansi  141 Oct  9 23:50 values-stage.yaml
-rw-rw-r-- 1 maansi maansi  29K Oct  9 23:54 superset-kube.yaml
-rw-rw-r-- 1 maansi maansi 8.2K Oct  9 23:57 final.yaml
-rwxrwxr-x 1 maansi maansi 3.2K Oct  9 23:58 pre-install-task.sh
drwxrwxr-x 2 maansi maansi 4.0K Oct  9 23:59 postgres
drwxrwxr-x 2 maansi maansi 4.0K Oct  9 23:59 superset
drwxrwxr-x 4 maansi maansi 4.0K Oct  9 23:59 redis
maansi@maansi-Standard-PC-Q35-ICH9-2009:~/superset-poc$
```

Step 19. Install podman.

```
sudo apt install podman
```

- sudo: Execute with superuser privileges.
- apt: Package management tool.
- install: Command to install a package.
- podman: Name of the package (a containerization tool).

Output:

Problem 3 I got error while installing podman.

```
mansi@mansi-Standard-PC-Q35-ICH9-2009:~/superset-poc$ sudo apt install podman
Reading package lists... Done
Building dependency tree
Reading state information... Done
E: Unable to locate package podman
```

For this i run below commands to solve this problem.

```
$ sudo sh -c "echo 'deb
https://download.opensuse.org/repositories/devel:/kubic:/libcontainers:/
stable/xUbuntu_$(lsb_release -rs)/ /' >
/etc/apt/sources.list.d/devel:kubic:libcontainers:stable.list"
```

```
$ wget
https://download.opensuse.org/repositories/devel:/kubic:/libcontainers:/
stable/xUbuntu_$(lsb_release -rs)/Release.key
```

```
$ sudo apt-key add - < Release.key
```

```
$ sudo apt update
```

```
$ sudo apt install -y podman
```

```
$ podman --version
```

"Note : Adding a repository is required for Ubuntu 20 version and is not needed for the latest version of Ubuntu."

Step 19. Create the pods and containers

```
# podman play kube final.yaml
```

- The command podman play kube final.yaml will read the Kubernetes YAML file final.yaml and recreate the pods and containers described in the file. create and start pods and containers without having to use the Kubernetes API or kubectl.

Output:

```
maansi@indianrenters-Latitude-5490:~/superset-poc$ podman play kube final.yaml
Trying to pull docker.io/bitnami/postgresql:14.6.0-debian-11-r13...
Getting image source signatures
Copying blob 380df3b45a2b done
Copying blob f8c1c832ce65 done
Copying config ff56fb78c6 done
Writing manifest to image destination
Storing signatures
Trying to pull docker.io/bitnami/redis:7.0.10-debian-11-r4...
Getting image source signatures
Copying blob 66e6a14f5a11 done
Copying config b762b5221e done
Writing manifest to image destination
Storing signatures
Resolved "apache/superset" as an alias (/home/maansi/.cache/containers/short-name-aliases.conf)
Trying to pull docker.io/apache/superset:dockerize...
Getting image source signatures
Copying blob 7264a8db6415 done
Copying blob f8213b5a0d73 done
Copying config 0066af9b14 done
Writing manifest to image destination
Storing signatures
Trying to pull apachesuperset.docker.scarf.sh/apache/superset:2.1.0...
Getting image source signatures
Copying blob 2de5f76669ff done
Copying blob 2d2b01660885 done
Copying blob f1f26f570256 done
Copying blob 2a34e3ad1a24 done
Copying blob 9450a9e89370 done
Copying blob 558c7c82036e done
Copying blob fd6e8aa0d19b done
Copying blob b8c33915aa76 done
Copying blob 4db7a27e56a7 done
Copying blob 1f3e6433dafd done
Copying blob f53268c71308 done
Copying blob c43b42a5c1bd done
Copying blob 31a5b66b66dc done
Copying blob dc53248fbae6 done
Copying blob db0542db1f78 done
Copying config 27d3d51cc6 done
Writing manifest to image destination
Storing signatures
Pod:
13d686ee154a1953c81ae1d94c76669a6ed0c28e584c1eb7bbee35e5150e005c
Container:
ea50f2829059eca40a6caced114f5fae5397d87030058165200300bf3625e452

Pod:
d362247981da2ddd81774c9c9d392b9ed421fb596b3766c75be1824de8db88aa
Container:
7d4a2828f343febb14d9c4cefbdb29ebc49f70a5a2812b691b4164967bfaa0df

Pod:
699341b47e5827c4409e833785a45d45da0ae7ed66dffa12093868531d070c9b
Container:
ab047f09c77ecd8f5cd7dbb4b5453eed32eedfe9e6d7b1a5d6d3f04e8d2721ea

Pod:
769d16c0f49af7dd5c594ea9df5dcbdf63af70710d8266235f96e399888e2be4
Container:
732561882f104314b6f4e45e45e735c6cc335960fad43a4cd7325fff8912c712

maansi@indianrenters-Latitude-5490:~/superset-poc$
```

Step 20. To see running containers.

```
# podman ps
```

- List the currently running containers managed by the Podman tool.
- Provides information about container IDs, names, status, ports, and more.

Output :

```
maansi@indianrenters-Latitude-5490:~/superset-poc$ podman ps
CONTAINER ID  IMAGE                                           COMMAND               CREATED            STATUS                       PORTS                   NAMES
c6977a321740  k8s.gcr.io/pause:3.5                                                  About an hour ago  Up About an hour ago                                 12b69f139b95-infra
34b796198509  docker.io/bitnami/postgresql:14.6.0-debian-11-r13  /opt/bitnami/scri...  About an hour ago  Up About an hour ago                                 kubepostgresql-pod-0-postgresql
5a8fdd0820ab  k8s.gcr.io/pause:3.5                                                  About an hour ago  Up About an hour ago         0.0.0.0:6379->6379/tcp  6cf5b91bcf86-infra
d205bb91b9a8  k8s.gcr.io/pause:3.5                                                  About an hour ago  Up About an hour ago                                 cb3482eeafb7-infra
5eefeeeeac48e apachesuperset.docker.scarf.sh/apache/superset:2.1.0                  About an hour ago  Up 15 minutes ago (healthy)                          kubesuperset-worker-pod-0-superset
e9553c90fe0d  k8s.gcr.io/pause:3.5                                                  About an hour ago  Up About an hour ago         0.0.0.0:8088->8088/tcp  0d2981fe9acf-infra
ea1b1e2bfb5e  apachesuperset.docker.scarf.sh/apache/superset:2.1.0                  About an hour ago  Up About an hour ago         0.0.0.0:8088->8088/tcp  kubesuperset-pod-0-superset
```

1) **Go to the browser and write :**

```
192.168.122.113:8088/login/
```

If you encounter error after this follow given below steps.

```
ls -lrth
```

```
aansi@indianrenters-Latitude-5490:~/superset-poc$ ls -lrth
otal 64K
rw-rw-r-- 1 maansi maansi  140 Oct  2 03:04 values-stage.yaml
rw-rw-r-- 1 maansi maansi  29K Oct  2 03:04 superset-kube.yaml
rwxrwxr-x 2 maansi maansi 4.0K Oct  2 03:08 superset
rwxrwxr-x 4 maansi maansi 4.0K Oct  2 03:09 redis
rwxrwxr-x 3 maansi maansi 4.0K Oct  2 03:09 postgres
rwxrwxr-x 1 maansi maansi 3.2K Oct  2 20:59 pre-install-task.sh
rw-rw-r-- 1 maansi maansi 8.3K Oct  3 11:05 final.yaml
aansi@indianrenters-Latitude-5490:~/superset-poc$
```

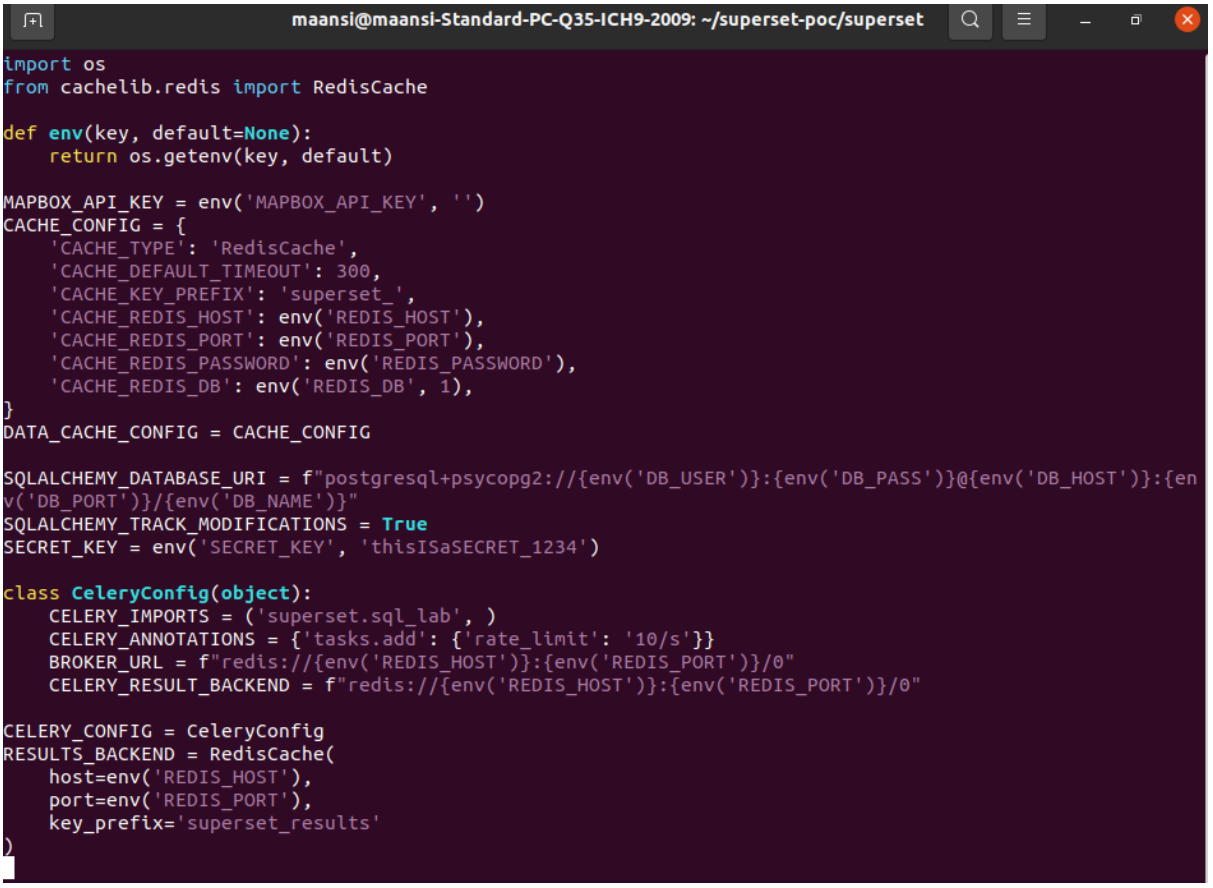
```
cd superset
```

```
maansi@indianrenters-Latitude-5490:~/superset-poc$ cd superset/
maansi@indianrenters-Latitude-5490:~/superset-poc/superset$
```

```
ls -lrth
```

```
maansi@indianrenters-Latitude-5490:~/superset-poc/superset$ ls -lrth
total 12K
-rwxrwxr-x 1 maansi maansi   96 Oct  2 03:08 superset_bootstrap.sh
-rwxrwxr-x 1 maansi maansi 1.2K Oct  2 03:08 superset_config.py
-rwxrwxr-x 1 maansi maansi  556 Oct  2 03:08 superset_init.sh
maansi@indianrenters-Latitude-5490:~/superset-poc/superset$
```

```
vim superset_config.py
```



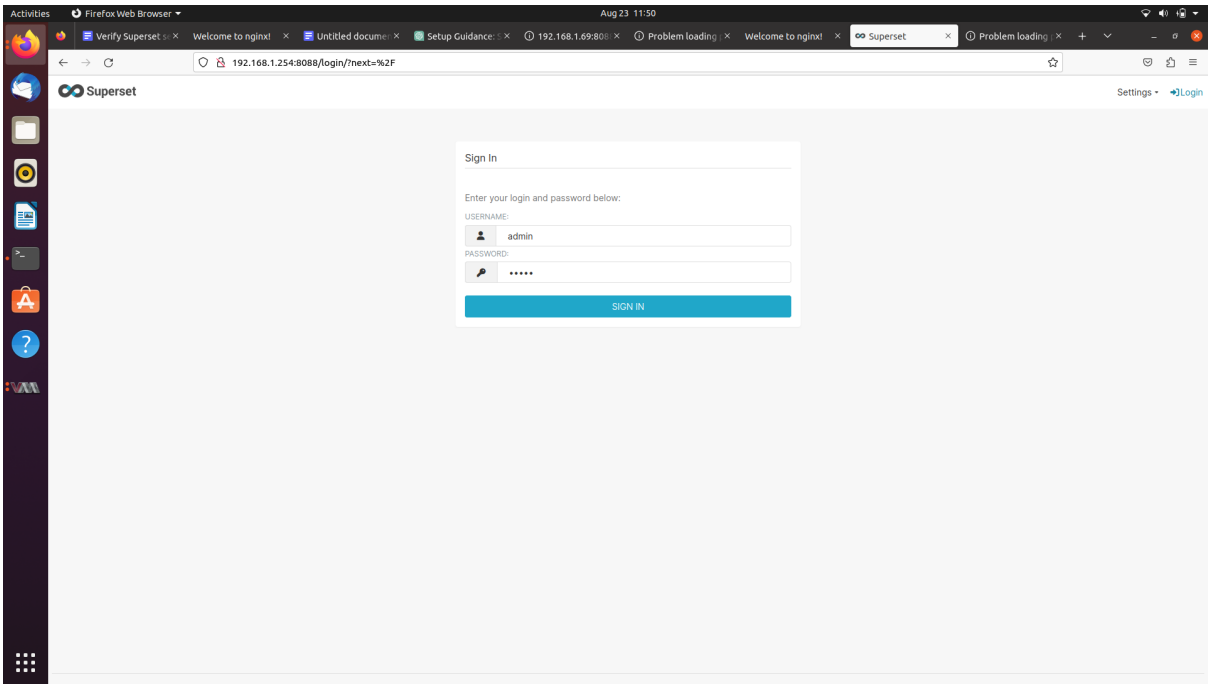
Note :- Correct its indentation and save it.

Test cases list

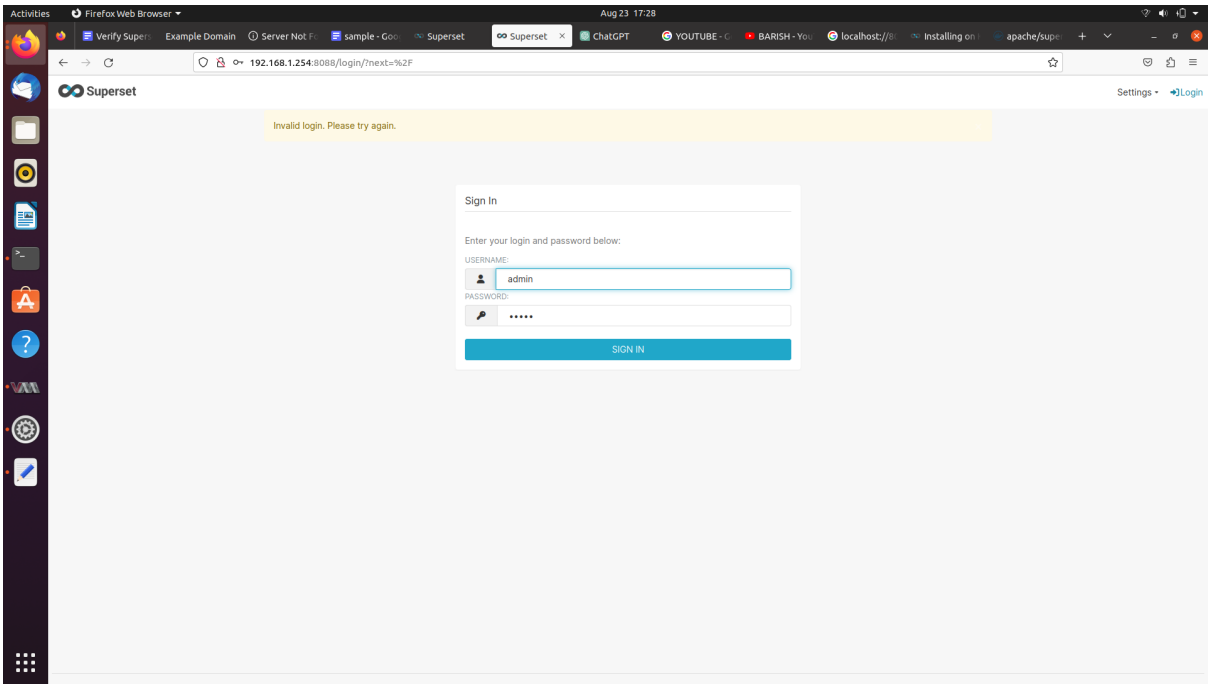
SNO	Component/Tool Name	Test case	Test Count	Test Cases	Expected Result	Test Passed [PASS/FAIL]	Remarks
a	Podman kube	Login to superset	1		Successful login	PASS	
b	Podman kube	Delete the pods and recreate			No data loss	PASS	

2) **Go to the browser and write :**

```
192.168.122.113:8088/login/
```



Problem 4 . When i go the browser and give password and username i got error.



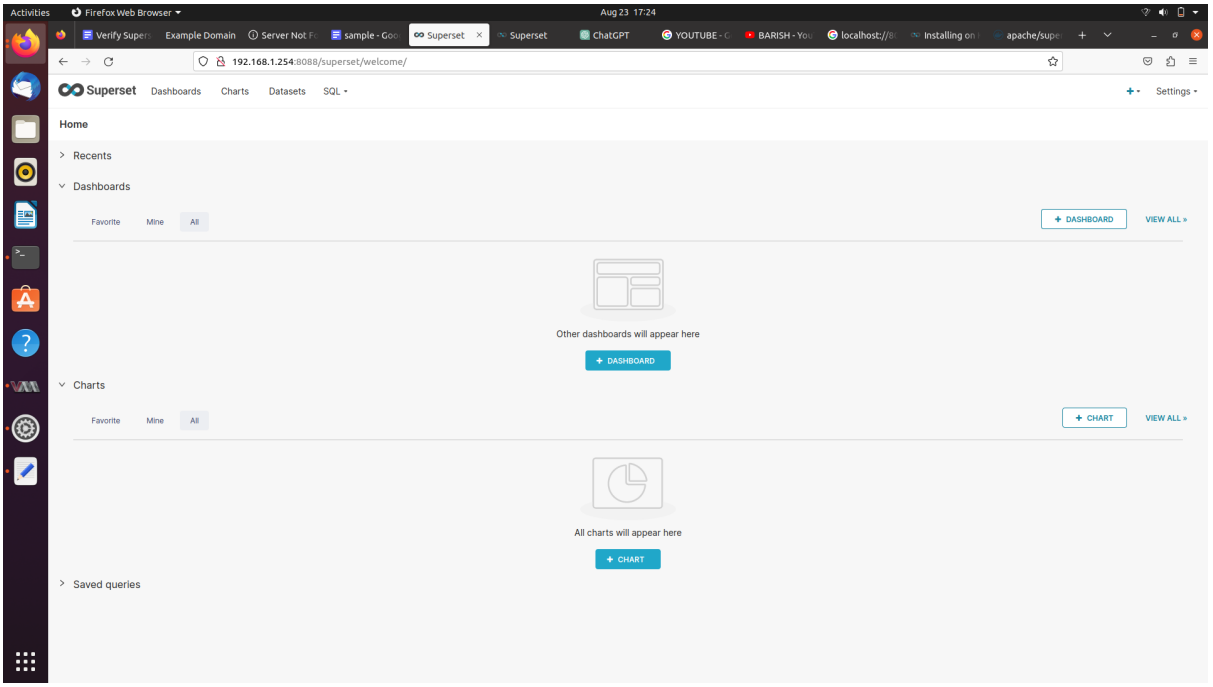
For this error i run below commands.

```
$ podman exec -it kubesuperset-pod-0-superset superset fab create-admin
```

```
$ podman exec -it kubesuperset-pod-0-superset superset db upgrade
```

```
$ podman exec -it kubesuperset-pod-0-superset superset load_examples
```

```
$ podman exec -it kubesuperset-pod-0-superset superset init
```



Reference link

https://helm.sh/docs/intro/install/

https://docs.podman.io/en/v4.2/markdown/podman-play-kube.1.html