Deployment Steps for Superset: -

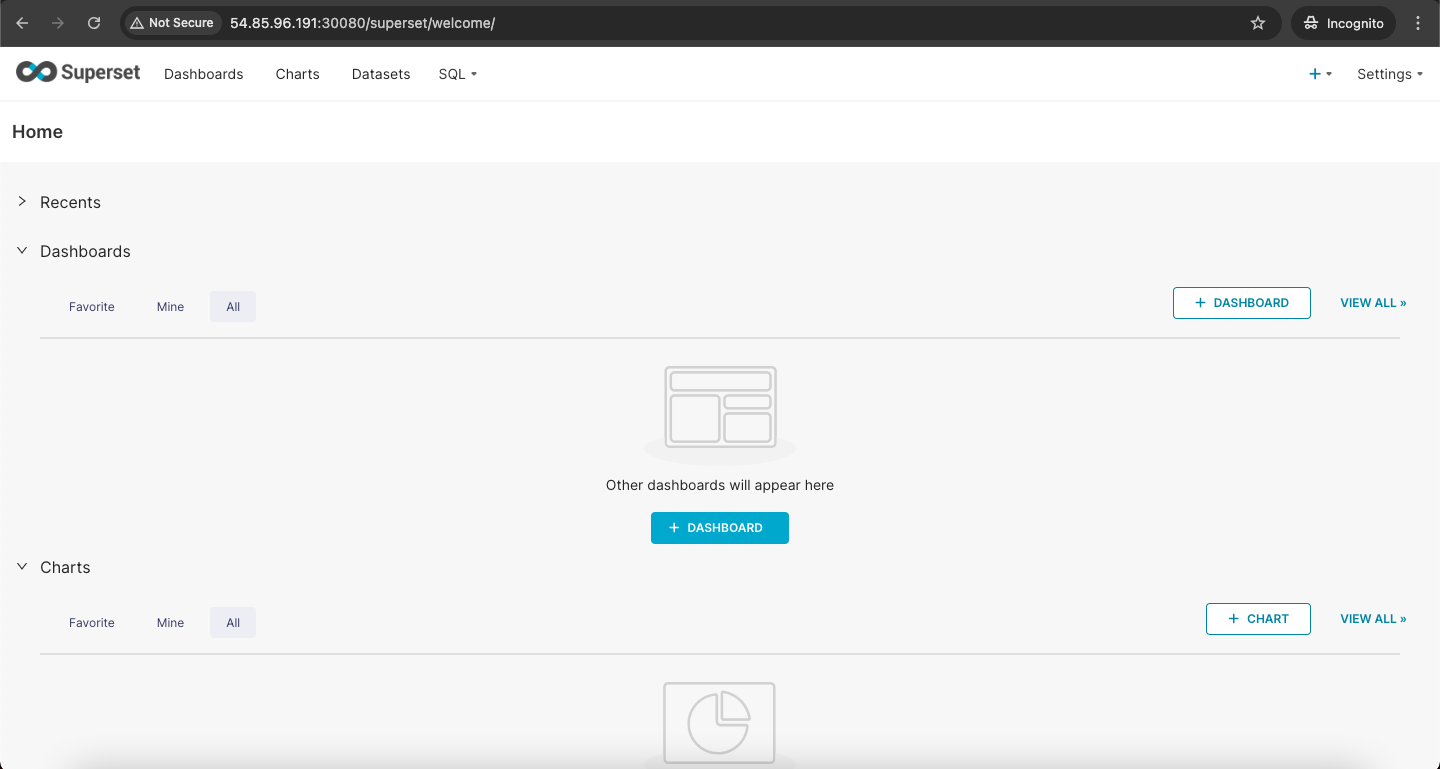
* Create a Persistent Volume[(pv)](https://github.com/shashank3656/DataZip/blob/master/superset/pv.yaml) (Note: - Using the Terraform Script it will automatically create the pv )
* Create a Persistent Volume Claim ([pvc](https://github.com/shashank3656/DataZip/blob/master/superset/pvc.yaml)) (Note: - Using the Terraform Script it will automatically create the pvc)
* Terraform Script ([Terraform Script](https://github.com/shashank3656/DataZip/tree/master/Terraform))

1. Cd Datazip/Terraform
2. Terraform init
3. Terraform validate
4. Terraform plan
5. Terraform apply

* Check the using below whether the superset is deployed in superset namespace

1. Kubectl get pv
2. Kubectl get pvc –n superset
3. Kubectl get pods –n superset
4. Kubectl get svc –n superset

URL: - http://<ipaddress>:<port>



Deployment Steps for Click House: -

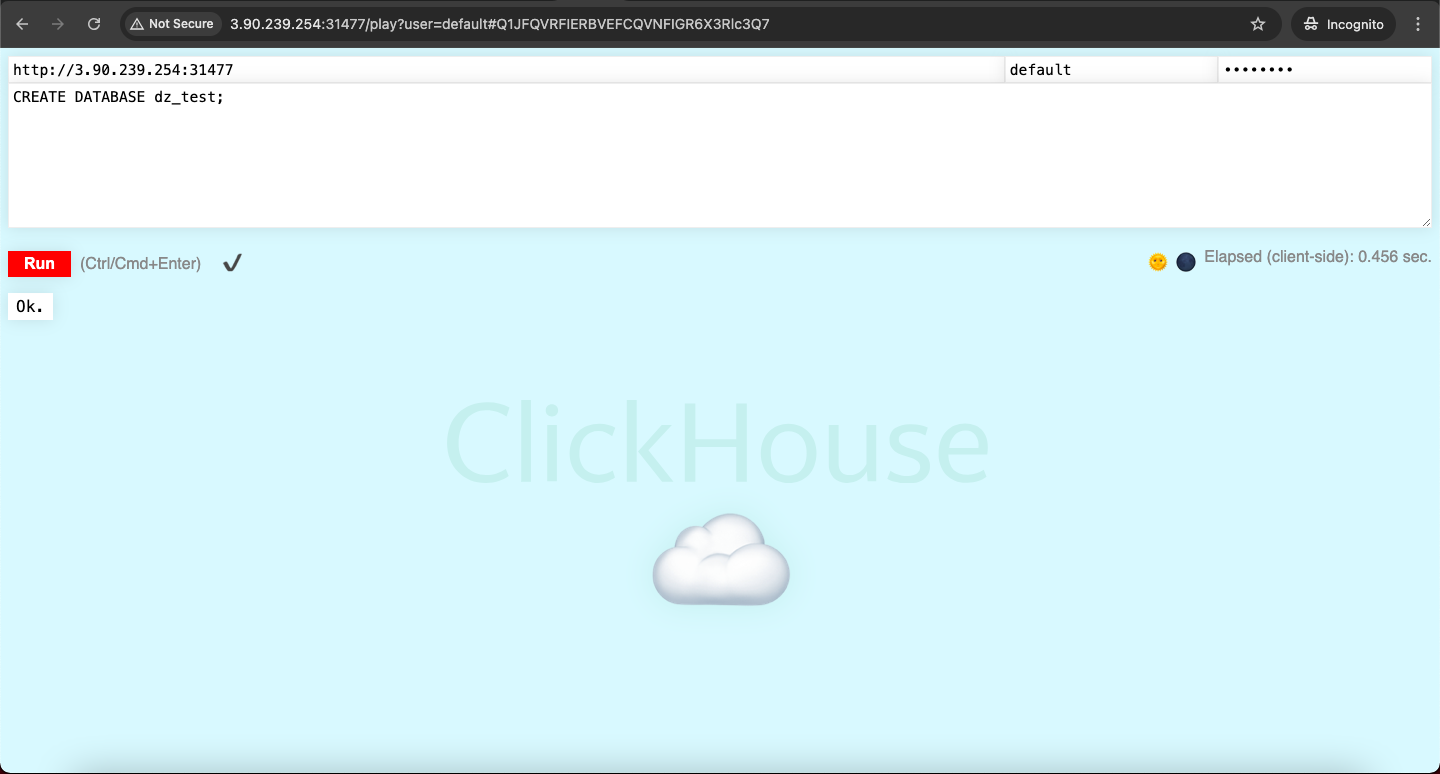
* Create a Persistent Volume[(pv)](https://github.com/shashank3656/DataZip/blob/master/Clickhouse/pv.yaml) (Note: - Using the Terraform Script it will automatically create the pv )
* Create a Persistent Volume Claim ([pvc](https://github.com/shashank3656/DataZip/blob/master/Clickhouse/pvc.yaml)) (Note: - Using the Terraform Script it will automatically create the pvc)
* Terraform Script ([Terraform Script](https://github.com/shashank3656/DataZip/tree/master/Terraform))

1. Cd Datazip/Terraform
2. Terraform init
3. Terraform validate
4. Terraform plan
5. Terraform apply

* Check the using below whether the superset is deployed in superset namespace

1. Kubectl get pv
2. Kubectl get pvc –n clickhouse
3. Kubectl get pods –n clickhouse
4. Kubectl get svc –n clickhouse

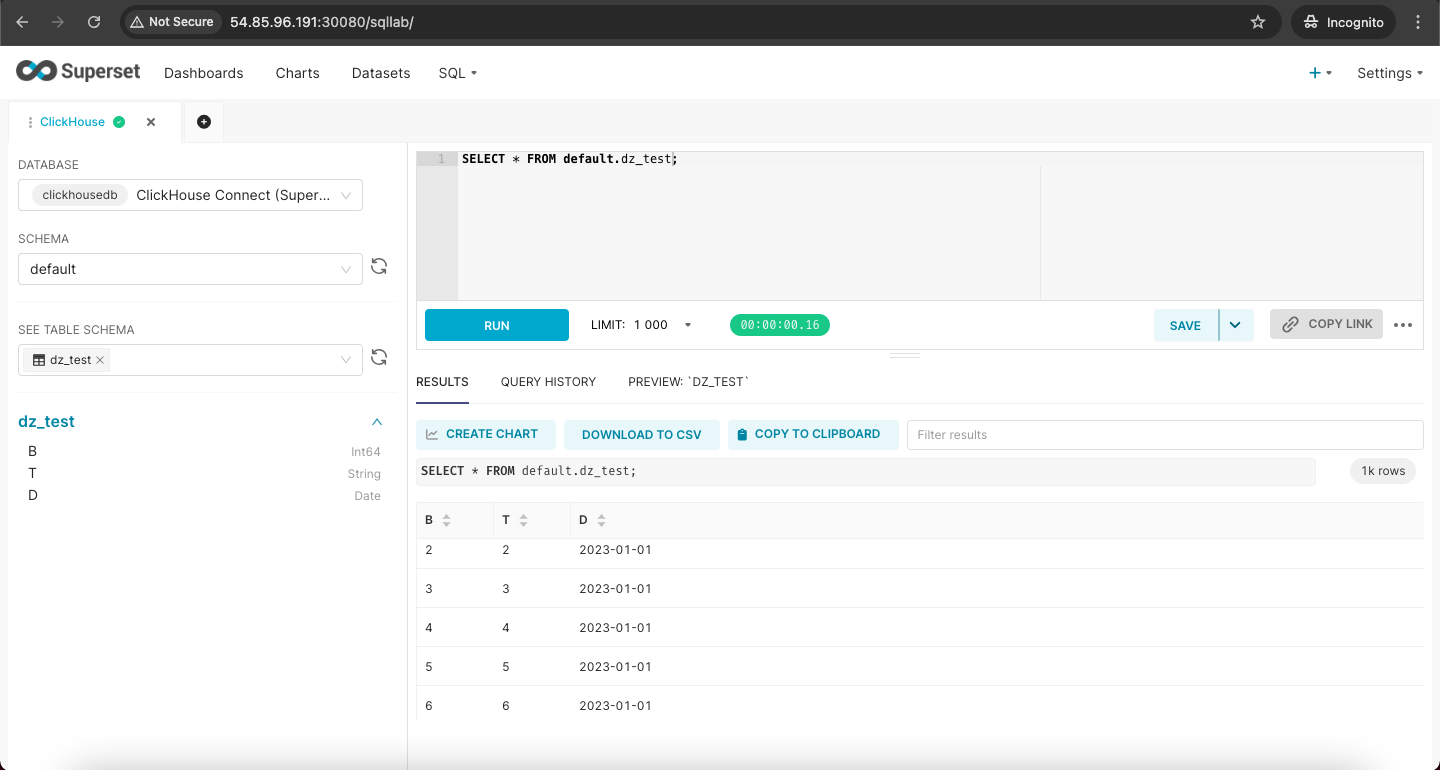
URL:- http://<ipaddress>:<port>/play



**Connection between the Superset and ClickHouse:-**

* Click the Database
* Click the Connect the database
* Select the ClickConnect
* Provide the Host
* Provide the port
* Provide the database
* Click Connect

Here the Query which connect to Click house and get Data



**Bonus Challenge (Not mandatory): Implement Hot-Cold Strategy**

* Install the S3 Driver in the Minikube for storing the cold data [(S3-Driver-Install)](https://github.com/shashank3656/DataZip/blob/master/Clickhouse/s3-driver.sh)

1. Wget <https://github.com/awslabs/mountpoint-s3-csi-driver/releases/download/helm-chart-aws-mountpoint-s3-csi-driver-1.11.0/aws-mountpoint-s3-csi-driver-1.11.0.tgz>
2. Tar zxvf aws-mountpoint-s3-csi-driver-1.11.0.tgz
3. cd aws-mountpoint-s3-csi-driver
4. helm install as-s3-driver . --values values.yaml -n kube-system --debug

* Create a Bucket in S3
* Create a iam role using and s3 bucket policy
* And the role to the VM’s as IAM Role
* Create a Persistent Volume & Persistent Volume Claim[(pv-s3)](https://github.com/shashank3656/DataZip/blob/master/Clickhouse/pv-s3.yaml)

Kubectl apply –f pv-s3.yaml

* Update the the Configmap [(ConfigMap-s3)](https://github.com/shashank3656/DataZip/blob/master/Clickhouse/config-s3.yaml)

Kubectl apply –f config-s3.yaml

* Deployment file for the [(deployemnt-s3)](https://github.com/shashank3656/DataZip/blob/master/Clickhouse/deployment.yaml)

Kubectl apply –f deployment-s3.yaml

