Superset K8s 部署

AUTHOR: 彭玲 TIME: 2022/11/16

背景

Superset 官方给出的 K8s 部署方式是通过 Helm 进行的，这种方式不便于维护。因此，使用官方提供的 superset 镜像进行部署，然而，该镜像不满足我们的实际需求，具体为：

- 默认使用语言为`英文`，期望汉化，使用`中文`。
- 默认不提供 ClickHouse 支持。

基于以上原因，我们基于 superset 镜像制作适合实际需要的镜像，并通过 K8s 方式部署。

镜像制作

安装 vim 工具

由于需要对 Superset 容器内容做修改，基于 superset 镜像，加入 vim 工具。

```
$ vi Dockerfile

FROM apache/superset:latest

USER root

RUN cp -a /etc/apt/sources.list /etc/apt/sources.list.bak; \
    sed -i 's/deb.debian.org/mirrors.ustc.edu.cn/g' /etc/apt/sources.list; \
    apt-get update; \
    apt-get install -y vim

USER superset
```

基于该 Dockerfile 构建镜像：

```
$ docker image build -t iot/superset .
```

设置中文 & 安装 ClickHouse 驱动

使用 `iot/superset` 镜像，启动容器（以`后台`方式运行）：

```
$ docker run -d -p 8080:8088 --name superset iot/superset
```

容器启动成功后，进入容器，并设置中文、安装 ClickHouse 驱动：

```
$ docker exec -it superset bash

# 设置中文
superset@10535a4cc4d9:/app$ vi superset/config.py

BABEL_DEFAULT_LOCALE = "zh"

superset@10535a4cc4d9:/app$ pybabel compile -d superset/translations

# 安装 ClickHouse 驱动
superset@10535a4cc4d9:/app$ pip install clickhouse-sqlalchemy==0.2.2
```

制作镜像

基于上面的 `superset` 容器制作镜像：

```
$ docker commit superset harbor.anxinyun.cn/iot/superset:22.11.16
```

镜像发布

登录 Harbor

使用 docker login 命令登录我司 Docker 私服：https://harbor.anxinyun.cn/ 。

```
$ docker login harbor.anxinyun.cn
Username: pengling
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```

Push 镜像到私服

将本地的 `harbor.anxinyun.cn/iot/superset:22.11.16` 镜像推送到 Harbor 私服。

```
$ docker push harbor.anxinyun.cn/iot/superset:22.11.16
The push refers to repository [harbor.anxinyun.cn/iot/superset]
541fae882580: Pushed
acb34a2b1633: Pushed
01898ba12ad4: Pushed
495b008e0a78: Pushed
e80e43843114: Pushed
a3b08391d898: Pushed
e55d798e9905: Pushed
f2f89bdb9ee1: Pushed
e65423f116f3: Pushed
253278beeb78: Pushed
0cdc6a6c6885: Pushed
a4041133a827: Pushed
600c4d5661a3: Pushed
8290d2857b36: Pushed
ea858b82d7e3: Pushed
630337cfb78d: Pushed
6485bed63627: Pushed
22.11.16: digest:
sha256:9c263ad468796d609309f0dde86cf4b6ccee28f96e069e1501aa2d8af5fbddf1 size: 3892
```

Superset 部署

yaml 部署文件

Deployment 部署文件如下：

```
$ vi deployment.yaml

apiVersion: apps/v1 # for versions before 1.8.0 use apps/v1beta1
kind: Deployment
metadata:
  name: superset-web
```

```
  namespace: julin
spec:
  selector:
    matchLabels:
      app: superset-web-deployment-template-label
  replicas: 1
  template:
    metadata:
      labels:
        app: superset-web-deployment-template-label
    spec:
      containers:
      - name: superset-web-deployment-container
        image: harbor.anxinyun.cn/iot/superset:22.11.16
        resources:
          requests:
            cpu: 100m
            memory: 100Mi
        ports:
        - containerPort: 8088 # 容器端口
```

Service 部署文件如下（ServiceType 根据实际情况确定）：

```
$ vi service.yaml

apiVersion: v1
kind: Service
metadata:
  name: superset-web
  namespace: julin
spec:
  type: NodePort
  ports:
  - port: 8080 # 外部端口
    targetPort: 8088 # 容器内端口
    nodePort: 30003
  selector:
    app: superset-web-deployment-template-label
```

基于这两个部署文件，创建 Superset 服务：

```
$ kubectl apply -f .
```

初始化 Superset 实例

Superset 容器运行后，进入容器，初始化 Superset 实例：

1. 设置 `admin` 账户

```
$ superset fab create-admin \
```

```
> --username admin \
> --firstname Superset \
> --lastname Admin \
> --email admin@superset.com \
> --password admin
--------------------------------------------------------------------------------
                                    WARNING
--------------------------------------------------------------------------------
A Default SECRET_KEY was detected, please use superset_config.py to override it.
Use a strong complex alphanumeric string and use a tool to help you generate
a sufficiently random sequence, ex: openssl rand -base64 42
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
logging was configured successfully
2022-11-16 01:32:25,859:INFO:superset.utils.logging_configurator:logging was
configured successfully
2022-11-16 01:32:25,865:INFO:root:Configured event logger of type <class
'superset.utils.log.DBEventLogger'>
We haven't found any Content Security Policy (CSP) defined in the configurations.
Please make sure to configure CSP using the TALISMAN_CONFIG key or any other
external software. Failing to configure CSP have serious security implications.
Check https://developer.mozilla.org/en-US/docs/Web/HTTP/CSP for more information.
You can disable this warning using the CONTENT_SECURITY_POLICY_WARNING key.
2022-11-16 01:32:25,866:WARNING:superset.initialization:We haven't found any
Content Security Policy (CSP) defined in the configurations. Please make sure to
configure CSP using the TALISMAN_CONFIG key or any other external software.
Failing to configure CSP have serious security implications. Check
https://developer.mozilla.org/en-US/docs/Web/HTTP/CSP for more information. You
can disable this warning using the CONTENT_SECURITY_POLICY_WARNING key.
Falling back to the built-in cache, that stores data in the metadata database,
for the following cache: `FILTER_STATE_CACHE_CONFIG`. It is recommended to use
`RedisCache`, `MemcachedCache` or another dedicated caching backend for
production deployments
2022-11-16 01:32:25,868:WARNING:superset.utils.cache_manager:Falling back to the
built-in cache, that stores data in the metadata database, for the following
cache: `FILTER_STATE_CACHE_CONFIG`. It is recommended to use `RedisCache`,
`MemcachedCache` or another dedicated caching backend for production deployments
Falling back to the built-in cache, that stores data in the metadata database,
for the following cache: `EXPLORE_FORM_DATA_CACHE_CONFIG`. It is recommended to
use `RedisCache`, `MemcachedCache` or another dedicated caching backend for
production deployments
2022-11-16 01:32:25,872:WARNING:superset.utils.cache_manager:Falling back to the
built-in cache, that stores data in the metadata database, for the following
cache: `EXPLORE_FORM_DATA_CACHE_CONFIG`. It is recommended to use `RedisCache`,
`MemcachedCache` or another dedicated caching backend for production deployments
```

```
/usr/local/lib/python3.8/site-packages/flask_appbuilder/models/sqla/interface.py:68: SAWarning: relationship 'SqlaTable.slices' will copy column tables.id to column slices.datasource_id, which conflicts with relationship(s): 'Slice.table' (copies tables.id to slices.datasource_id). If this is not the intention, consider if these relationships should be linked with back_populates, or if viewonly=True should be applied to one or more if they are read-only. For the less common case that foreign key constraints are partially overlapping, the orm.foreign() annotation can be used to isolate the columns that should be written towards.   To silence this warning, add the parameter 'overlaps="table"' to the 'SqlaTable.slices' relationship. (Background on this error at: https://sqlalche.me/e/14/qzyx)
  for prop in class_mapper(obj).iterate_properties:
Recognized Database Authentications.
Admin User admin created.
```

2. 升级数据库

```
$ superset db upgrade
```

3. 加载示例（该步骤**可省略**）

```
$ superset load_examples
```

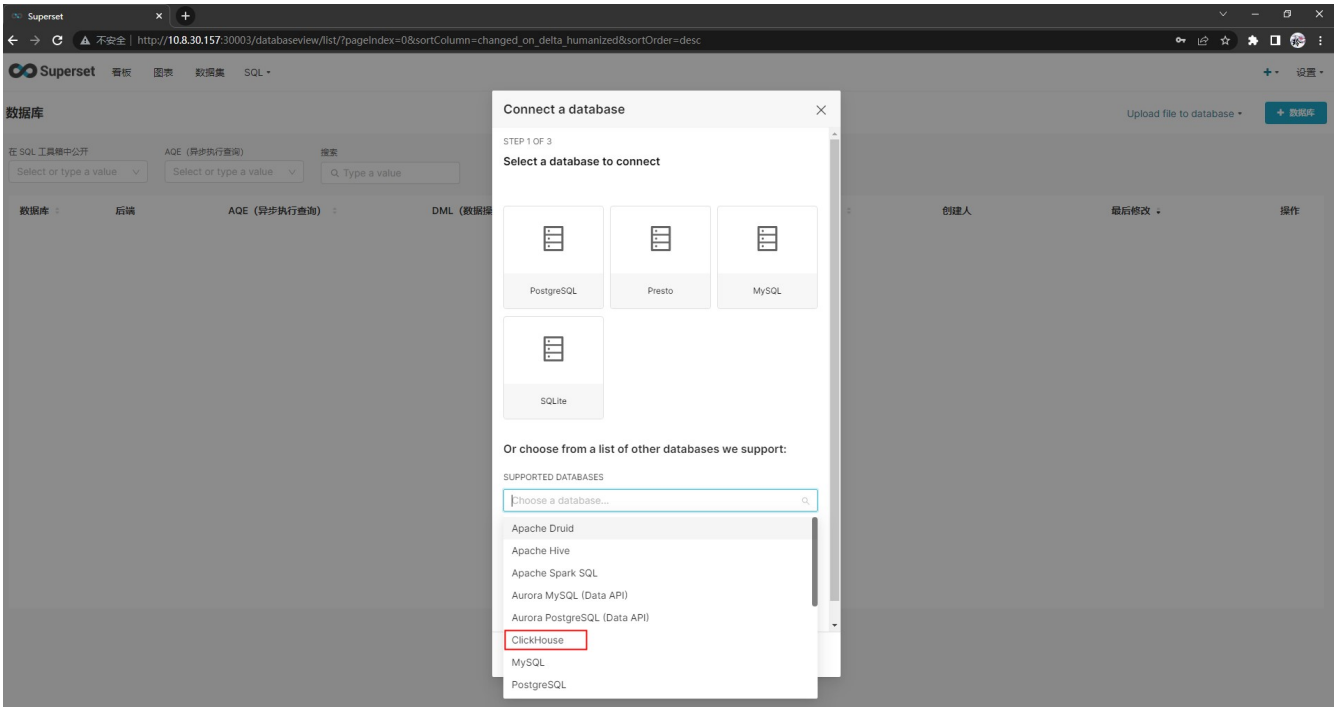
4. 设置角色

```
$ superset init
```

Superset 网站访问

至此，我们可以正常访问 Superset 网站：http://10.8.30.157:30003/ 。登录用户名/密码：`admin`/`admin`。

进入`数据库连接`页面，新增数据库，此时，支持的数据库列表中包含了`ClickHouse`选项（未装 clickhouse 驱动前，无此选项）：


Superset
不安全 | http://10.8.30.157:30003/databaseview/list/?pageIndex=0&sortColumn=changed_on_delta_humanized&sortOrder=desc
Superset 看板 图表 数据集 SQL
设置
数据库
Upload file to database
+ 数据库
在 SQL 工具箱中公开
AQE（异步执行查询）
搜索
Select or type a value
Type a value
数据库
后端
AQE（异步执行查询）
DML（数据操
创建人
最后修改
操作
Connect a database
STEP 1 OF 3
Select a database to connect
PostgreSQL
Presto
MySQL
SQLite
Or choose from a list of other databases we support:
SUPPORTED DATABASES
Choose a database...
Apache Druid
Apache Hive
Apache Spark SQL
Aurora MySQL (Data API)
Aurora PostgreSQL (Data API)
ClickHouse
MySQL
PostgreSQL