Apache Airflow

Airflow

1. Configuration Files Overview

   All configuration files for the Airflow deployment can be found in the directory /root/cafebot-kube/airflow.

2. Important Files

   a. **/root/cafebot-kube/airflow/values.yml:** This file is located within the Helm chart and contains default configuration values for the chart templates. It allows users to customize the behavior of the chart without directly modifying the templates.

   b. **/root/cafebot-kube/airflow/airflowScaledObject.yml:** This file specifies configurations for scaling Airflow and includes details such as trigger authentication and scaling criteria.

3. Configuration Details

   All configuration for a Helm chart is controlled by the **values.yml** file located within the Helm chart. This file contains default configuration values for the chart templates and allows users to customize the behavior of the chart without directly modifying the templates.

   1. Airflow Image Configuration (Airflow.image)

Defines configuration for the Airflow image, including built-in DAGs.

```
airflow:
   ## if we use legacy 1.10 airflow commands
   ##
   legacyCommands: false

   ## configs for the airflow container image
   ##
   image:
     repository: shekharzxcv/airflow
     tag: 2.6.3-python3.9
     pullPolicy: IfNotPresent
     pullSecret: ""
     uid: 50000
     gid: 0
```

2. User Configuration (users)

```
users:
     - username: admin
       password: admin
       role: Admin
       email: admin@example.com
       firstName: admin
       lastName: admin
```

3. External Database Configuration (External Database

```
externalDatase:
     type: postgres
     host:
     port:
     database:
     user:
     password:
```

4. Including DAGs in the Image

   a. Place DAGs in the directory: /root/cafebot-kube/airflow/dags.

   b. Define requirements in /root/cafebot-kube/airflow/requirements.txt.

   Build the Airflow image.

```
cd /root/cafebot-kube/airflow
 docker build . -t repo:tag
 docker push repo:tag
```

```
FROM apache/airflow:2.6.3-python3.9

 USER root

 COPY --chown=airflow:root dags/ /opt/airflow/dags
 USER airflow
```

   Example:

```
docker build . -t shekharzxcv/airflow:2.6.3-python3.9
 docker push shekharzxcv/airflow:2.6.3-python3.9
```

5. Helm chart upgrade

   After making any changes in the configuration, you need to upgrade the existing release with the updated configuration.

   - To list the helm releases:

```
root@kube-dev:~# helm list -n cafebot2
 NAME                            NAMESPACE       REVISI
 airflow                         cafebot2        1
 clickhouse                      cafebot2        1
 haproxy-kubernetes-ingress      cafebot2        63
 keda                            cafebot2        1
 local-path-provisioner          cafebot2        1
```

```
metrics-server                  cafebot2        1
nifi-nifi                       cafebot2        1
prometheus                      cafebot2        12
root@kube-dev:~#
```

- TO upgrade the helm release

```
helm upgrade airflow . -n cafebot2 -f values.yml
```

Airflow Scaled Object

The `ScaledObject` is used by KEDA to define scaling behavior for an application. It sets the minimum and maximum number of replicas and defines triggers for scaling.

The `TriggerAuthentication` provides authentication and connection details for the trigger used by KEDA. In this case, it authenticates and connects to a PostgreSQL database.

By using these resources, KEDA can dynamically scale the `airflow-worker` based on the specified trigger conditions.

airflowScaledObject.yml

```
apiVersion: keda.sh/v1alpha1
kind: TriggerAuthentication
metadata:
  name: cafebot-pg-auth  # Replace with your desired n
  namespace: cafebot2
spec:
  secretTargetRef:
    - parameter: dbname  # Env variable name for datab
      name: spark  # Secret name changed to "flyingspa
      key: dbname  # Key within the Secret
    - parameter: password  # Env variable name for pas
      name: spark
      key: password
    - parameter: user  # Env variable name for usernam
```

```
      name: spark
      key: user

---
apiVersion: keda.sh/v1alpha1
kind: ScaledObject
metadata:
  name: airflow-worker
  namespace: cafebot2
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: StatefulSet
    name: airflow-worker
  pollingInterval: 10
  cooldownPeriod:  20
  minReplicaCount: 1
  maxReplicaCount: 10
  triggers:
    - type: postgresql
      metadata:
        userName: spark
        host: cafebot-pgbouncer.cafebot2.svc.cluster.l
        port: "5432"
        dbName: spark
        passwordFromEnv: password
        sslmode: require
        query: "SELECT ceil(COUNT(*)::decimal / 16) FR
        targetQueryValue: "2.2"
      authenticationRef:
        name: cafebot-pg-auth
```

This file contains two configurations:

▼ Trigger Authentication

Defines PostgreSQL database credentials from the secret "spark." Update the secret name if the PostgreSQL credentials change.

▼ Scaling Criteria

Specifies the trigger query for scaling Airflow.

Applying Changes:

```
kubectl apply -f airflowScaledObject.yml
```

Listing ScaledObjects

```
kubectl get scaledobject -n cafebot2
```

```
example:
 root@kube-dev:~# kubectl get scaledobject -n cafebo
 NAME                     SCALETARGETKIND      SCAL
 airflow-worker           apps/v1.StatefulSet   airf
 cafebot-instance1-fld5   apps/v1.StatefulSet   cafe
 cafebot-instance1-kndx   apps/v1.StatefulSet   cafe
 cafebot-instance1-svt2   apps/v1.StatefulSet   cafe
 nifi-scaled              apps/v1.StatefulSet   nifi
 pyspark-pi                                     pysp
 spark-scaled             apps/v1.StatefulSet   spar
 root@kube-dev:~#
```

Displays scaled object details, including the object's name, deployment, statefulset, max and min replica counts, and the configured scaler.

This documentation provides comprehensive insights into Airflow configuration, image building, and scaling procedures. Ensure to follow these steps meticulously for a successful Airflow deployment.