Apache NiFi Documentation

1. Path to configurations files

   All NiFi configurations are located in the directory: /root/cafebot-kube/nifi.

2. Building Custom Nifi Image

   To build a custom NiFi image with custom JAR libraries, use the following Dockerfile. Make sure to place all custom JARs in /root/cafebot-kube/nifi/jars before building and pushing the image.

   > 💡 Navigate to the directory "/root/cafebot-kube/nifi" that contains the Dockerfile.

   - Build and push the image

   ```
   docker build . -t repo:tag
   docker push repo:tag
   ```

   ```
   #Dockerfile
   FROM apache/nifi:1.23.2
   ```

```
USER 0

# Create the directory to house the JAR files in the speci
RUN mkdir -p /opt/configuration_resources/custom_lib

# Copy all JAR files from the 'jars' directory into the cr
COPY jars /opt/configuration_resources/custom_lib/

# Change ownership of the custom_lib directory to the 'nif
RUN chown -R nifi:nifi /opt/configuration_resources/custom

USER nifi
```

In this case, the tag is shekharzxcv/nifi:jars.

After pushing the image to the remote repository, update the image in the values.yml file of the NiFi Helm chart.

3. Nifi Configuration.

   Path: /root/cafebot-kube/nifi/values.yml

   Update the image name in values.yml as follows

```
image:
  repository: shekharzxcv/nifi
  tag: "jars"
  pullPolicy: "IfNotPresent"
```

   and the upgrade the nifi helm release with the updated configuration.

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

4. Scaling Nifi Based upon metrics

   To scale NiFi based on metrics, enable metrics in the metrics section of values.yml.

```
metrics:
  enabled: true
```

5. Nifi ScaledObject Configuration

The nifiScaledObject.yml file is the ScaledObject configuration for NiFi. This enables scaling NiFi based on the queue in NiFi. Configure the query and threshold in this file.

After making changes in nifiScaledObject.yml , apply these changes.

```
kubectl apply -f path-of-the-file
 kubectl apply -f nifiScaledObject.yml
```

ScaledObject.yml

```
apiVersion: keda.sh/v1alpha1
 kind: ScaledObject
 metadata:
   name: nifi-scaled
   namespace: cafebot2
 spec:
   scaleTargetRef:
     apiVersion: apps/v1
     kind: StatefulSet
     name: nifi-nifi
   pollingInterval: 10
   cooldownPeriod: 20
   minReplicaCount: 1
   maxReplicaCount: 10
   triggers:
     - type: prometheus
       metadata:
         serverAddress: "http://10.245.12.18:80"
         query: "max(nifi_amount_items_queued) > 50"
```

```
        threshold: '100'  # Adjust the threshold value bas
        activationThreshold: '1'
```

6. Listing ScaledObject

```
example:
 root@kube-dev:~# kubectl get scaledobject -n cafebot2
 NAME                     SCALETARGETKIND      SCALETARGET
 airflow-worker           apps/v1.StatefulSet   airflow-wor
 cafebot-instance1-fld5   apps/v1.StatefulSet   cafebot-ins
 cafebot-instance1-kndx   apps/v1.StatefulSet   cafebot-ins
 cafebot-instance1-svt2   apps/v1.StatefulSet   cafebot-ins
 nifi-scaled              apps/v1.StatefulSet   nifi-nifi
 pyspark-pi                                     pyspark-pi
 spark-scaled             apps/v1.StatefulSet   spark-worke
 root@kube-dev:~#
```

This section displays the ScaledObject name, the deployment and statefulset it is attached to, the maximum and minimum replica count, and the configured scaler. It also provides the current state of the ScaledObject.

This documentation comprehensively guides through NiFi configuration, image building, and scaling procedures. Follow these steps meticulously for a successful NiFi deployment.