The demo app written in go needs kafka and redis.The steps for this demo app deploy is on a amazon ec2 machine (commands are similar to centos).

Do a update on a fresh instance.

#yum update

Install necessary packages

#yum install epel-release vim git wget curl gcc gcc-c++

Install Redis

#yum install redis

Start redis service

#service redis start

Enable redis on-boot

#systemctl enable redis

Check redis status

#service redis status

check redis working

# redis-cli

127.0.0.1:6379> INFO

# Server

redis\_version:3.2.12

redis\_git\_sha1:00000000

.... output emitted...

Install java for kafka and zookeeper

#yum install java-1.8.0-openjdk

#java -version

Download and install kafka and zookeeper

# cd /opt/ && wget https://mirror.ibcp.fr/pub/apache/kafka/2.5.0/kafka\_2.12-2.5.0.tgz

# tar -xzf kafka\_2.12-2.5.0.tgz

# mv kafka\_2.12-2.5.0 kafka

#vim /lib/systemd/system/zookeeper.service

COpy paste the below in the above file.

[Unit]

Description=Zookeper daemon

After=network.target

[Service]

Type=simple

ExecStart=/opt/kafka/bin/zookeeper-server-start.sh -daemon /opt/kafka/config/zookeeper.properties

KillMode=process

Restart=on-failure

RestartSec=42s

[Install]

WantedBy=multi-user.target

start and enable zookeeper.

# systemctl enable zookeeper.service

# systemctl start zookeeper.service

# systemctl status zookeeper.service

Add kafka systemd service.

vim /lib/systemd/system/kafka.service

copy paste the below contenet in the above file.

[Unit]

Description=Kafka daemon

After=zookeper.service

[Service]

Type=simple

ExecStart=/opt/kafka/bin/kafka-server-start.sh -daemon /opt/kafka/config/server.properties

KillMode=process

Restart=on-failure

RestartSec=42s

[Install]

WantedBy=multi-user.target

before starting Kafka service we need to put these lines at the end of server.properties file:

listeners=PLAINTEXT://:9092

num.network.threads=3

num.io.threads=8

socket.send.buffer.bytes=102400

socket.receive.buffer.bytes=102400

socket.request.max.bytes=104857600

# systemctl enable kafka.service

# systemctl start kafka.service

# systemctl status kafka.service

# cd /opt/ && wget https://github.com/SourceLabOrg/kafka-webview/releases/download/v2.5.1/kafka-webview-ui-2.5.1-bin.zip

# unzip kafka-webview-ui-2.5.1-bin.zip

# mv kafka-webview-ui-2.5.1-bin kafka-webview

#vim /lib/systemd/system/kafka-webview.service

copy paste the below entry in the above file.

[Unit]

Description=Kafka Web UI daemon

After=kafka.service

[Service]

Type=simple

ExecStart=/opt/kafka-webview/start.sh

KillMode=process

Restart=on-failure

RestartSec=42s

[Install]

WantedBy=multi-user.target

start the kafka-webview service.

# systemctl enable kafka-webview.service

# systemctl start kafka-webview.service

# systemctl status kafka-webview.service

Install Iptabels and stop firewalld

#yum install iptables-services.

#systemctl stop firewalld

#systemctl disable firewalld

#systemctl mask firewalld

#service start iptables

Add iptables rule for app and kafka webview and ssh

#iptables -A INPUT -p tcp --dport ssh -j ACCEPT

#iptables -A INPUT -p tcp --dport 8080 -j ACCEPT

#iptables -A INPUT -p tcp --dport 8081 -j ACCEPT

Install go:

#wget https://golang.org/dl/go1.15.6.linux-amd64.tar.gz

#tar -C /usr/local -xzf go1.15.6.linux-amd64.tar.gz

#export PATH=$PATH:/usr/local/go/bin

clone the repo from git.

#cd /opt

#git clone git@gitlab.com:melwyn95/go-redis-kafka-demo.git

Create a topic called 'numbers' using kafka-webview (open using <ip>:8080.you can skip authentication in the kafka-webview config yaml file). First you need to add the kafka cluster in it.

#mv go-redis-kafka-demo-master app

cd into /opt/app

modify the start.sh as below.

\_\_

#!/bin/bash

export REDIS\_HOST="localhost";

export REDIS\_PORT="6379";

export REDIS\_PASSWORD="";

export KAFKA\_HOST="localhost";

export KAFKA\_PORT="9092";

export KAFKA\_TOPIC="number";

export KAFKA\_CONSUMER\_GROUP="numbers-group";

export HTTP\_SERVER\_PORT="8081";

export HTTP\_SERVER\_TIMEOUT="10000";

export GO111MODULE=on;

go mod download

go install ./...

go run cmd/http-server/main.go

\_\_\_\_\_\_

run the start.sh file in background.

sh start.sh &