\*\*Flink Autoscaler with EMR on EKS\*\*

sudo yum install -y bash-completion jq

1. Manually create a "provisioned" MSK cluster on AWS console. Make sure it's in the same VPC as our EKS cluster. ( put it into a lab’s CFN)

2. Setup your env variables and ensure you can connect to the kafka cluster:

```

NAMESPACE=emr-flink

EMR\_VERSION=7.2.0

AWS\_REGION=us-west-2

BUCKET\_NAME=my-demo-bucket

CHECKPOINT\_S3\_STORAGE\_PATH=s3://${BUCKET\_NAME}/flink/autoscaling/checkpoint/

SAVEPOINT\_S3\_STORAGE\_PATH=s3://${BUCKET\_NAME}/autoscaling/savepoint/

HIGH\_AVAILABILITY\_STORAGE\_PATH=s3://${BUCKET\_NAME}/flink/autoscaling/ha/

EMR\_EXECUTION\_ROLE\_ARN=$(aws iam get-role --role-name emr-on-eks-test-execution-role| jq -r .Role.Arn)

kubectl config set-context --current --namespace $NAMESPACE

JOB\_SCRIPT\_FILE=s3://${BUCKET\_NAME}/flink/script/datagen-kafka-script.py

JOB\_SCRIPT\_FILE\_NAME=datagen-kafka-script.py

ORDER\_DATA\_TOPIC\_NAME="order-data"

ORDER\_DATA\_AGG\_TOPIC\_NAME="order-data-agg"

# Note that this is the first MSK cluster. If you have multiple MSK clusters, please find the cluster ARN you need. Use the account provided by the Workshop. There will only be one MSK cluster. Just execute the following command

MSK\_ARN=$(aws kafka list-clusters |jq -r .ClusterInfoList[0].ClusterArn)

MSK\_BROKER=$(aws kafka get-bootstrap-brokers --cluster-arn $MSK\_ARN |jq -r .BootstrapBrokerString)

echo -e " $NAMESPACE \n $BUCKET\_NAME \n $CHECKPOINT\_S3\_STORAGE\_PATH \n $SAVEPOINT\_S3\_STORAGE\_PATH \n $HIGH\_AVAILABILITY\_STORAGE\_PATH \n $EMR\_EXECUTION\_ROLE\_ARN \n $JOB\_SCRIPT\_FILE \n $JOB\_SCRIPT\_FILE\_NAME \n $MSK\_BROKER \n $ORDER\_DATA\_TOPIC\_NAME \n$ORDER\_DATA\_AGG\_TOPIC\_NAME \n $EMR\_VERSION"

```

3. install the latest Flink Operator ( emr7.2) in your EKS cluster

4. Install kafka client tool in your terminal:

```

wget https://archive.apache.org/dist/kafka/2.8.1/kafka\_2.12-2.8.1.tgz .

tar -xvzf kafka\_2.12-2.8.1.tgz

cd kafka\_2.12-2.8.1

```

5. Create 2 empty kafka topics:

```

# Source Topic : 12 partition

./bin/kafka-topics.sh --bootstrap-server $MSK\_BROKER --create --replication-factor 2 --partitions 12 --topic $ORDER\_DATA\_TOPIC\_NAME

./bin/kafka-topics.sh --bootstrap-server $MSK\_BROKER --topic $ORDER\_DATA\_TOPIC\_NAME --describe

# Output Topic : 6 partition

./bin/kafka-topics.sh --bootstrap-server $MSK\_BROKER --create --replication-factor 2 --partitions 6 --topic $ORDER\_DATA\_AGG\_TOPIC\_NAME

./bin/kafka-topics.sh --bootstrap-server $MSK\_BROKER --topic $ORDER\_DATA\_AGG\_TOPIC\_NAME --describe

```

6. The job need EBS and Karpenter, so install an EBS CSI Driver add-on and karpenter:

```

eksctl create addon \

--name aws-ebs-csi-driver \

--region $AWS\_REGION \

--cluster $EKS\_CLUSTER \

--service-account-role-arn arn:aws:iam::$ACCOUNT:role/AmazonEKS\_EBS\_CSI\_DriverRole

cat << EOF -> storage-class.yaml

apiVersion: storage.k8s.io/v1

kind: StorageClass

metadata:

name: ebs-sc

provisioner: ebs.csi.aws.com

volumeBindingMode: WaitForFirstConsumer

EOF

kubectl apply -f storage-class.yaml

kubectl get sc

```

```

# update the script to match your own EKS version.

Curl -O https://raw.githubusercontent.com/hitsub2/flink-on-eks/main/karpenter.sh

chmod +x karpenter.sh

bash karpenter.sh

7. Build 2 docker images ( by the way, EMR 7.8 only allows to build multi-platform image, why???)

```

#login to ecr

aws ecr get-login-password --region ${AWS\_REGION} |docker login --username AWS --password-stdin ${ACCOUNT}.dkr.ecr.${AWS\_REGION}.amazonaws.com

# create ecr repos

REPO\_NAME=emr-7.8.0-flink-datgen-msk

aws ecr create-repository \

--repository-name ${REPO\_NAME} \

--image-scanning-configuration scanOnPush=true \

--region ${AWS\_REGION}

REPO\_NAME=emr-7.8.0-flink-msk-s3

aws ecr create-repository \

--repository-name ${REPO\_NAME} \

--image-scanning-configuration scanOnPush=true \

--region ${AWS\_REGION}

# build and push

docker buildx build --platform linux/amd64,linux/arm64 -t $ACCOUNT.dkr.ecr.$AWS\_REGION.amazonaws.com/emr-7.2.0-flink-datgen-msk:latest -f Dockerfile-datgen --build-arg EMR\_VERSION=${EMR\_VERSION} --push .

docker buildx build --platform linux/amd64,linux/arm64 -t $ACCOUNT.dkr.ecr.$AWS\_REGION.amazonaws.com/emr-7.2.0-flink-msk-s3 -f Dockerfile-msk-s3 --build-arg EMR\_VERSION=${EMR\_VERSION} --push .

```

8. Upload 2 pyFlink apps scripts to s3:

```

aws s3 cp kafka-s3-sink.py s3://$BUCKET\_NAME/flink/script/kafka-s3-sink.py

aws s3 cp datagen-kafka-script.py s3://$BUCKET\_NAME/flink/script/datagen-kafka-script.py

```

9. Get the flink operator job yamls from attachment, update some attributes based on your environment, then run:

```

# generate source data to kafaka

kubectl apply -f flink-datagen-kafka.yaml -n YOUR\_NAMESPACE

# consume the source and produce aggregated data to s3 and kafka

# this job will failed with EMR 7.8, but works on EMR 7.2

kubectl apply -f flink-kafka-s3.yaml -n YOUR\_NAMESPACE

```

10. Validate in Kafka:

```

# source topic

bin/kafka-console-consumer.sh --bootstrap-server $MSK\_BROKER --topic order-data

#output topic

bin/kafka-console-consumer.sh --bootstrap-server $MSK\_BROKER --topic order-data-agg

```