Setup performance metrics collection for PM Mapper

Pre-requisites

- Single Node Kubernetes cluster environment (e.g. microk8s)
- Prometheus service up and running (https://linuxacademy.com/blog/kubernetes/running-prometheus-on-kubernetes/)
- Grafana dashboard up and running
    - Use your Prometheus service as datasource
    - Create Dashboard before hand e.g. Kubernetes Pod Metrics rev 2-1562689587039.json

trust.pass

trust.jks.b64

jks.pass

cert.jks.b64

pmm.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: pmmapper
spec:
  selector:
    matchLabels:
      run: pmmapper
  replicas: 1
  template:
    metadata:
      labels:
        run: pmmapper
    spec:
      containers:
      - name: pmmapper
        image: "nexus3.onap.org:10001/onap/org.onap.dcaegen2.services.pm-mapper:<CHANGE THIS TO YOUR VERSION>"
        env:
        - name: CONFIG_BINDING_SERVICE_SERVICE_HOST
          value: $(SIMULATOR_SERVICE_HOST)
        - name: CONFIG_BINDING_SERVICE_SERVICE_PORT
          value: $(SIMULATOR_SERVICE_PORT)
        - name: HOSTNAME
          value: "pm-mapper"
        ports:
        - containerPort: 8081
        volumeMounts:
        - name: config-volume
          mountPath: /opt/app/pm-mapper/etc/cert/PM Mapper initial performance metrics
      volumes:
      - name: config-volume
        configMap:
          name: pmmapper-config
```

metadata.json

```
{"productName": "ACME_Node","vendorName": "ACME Corp","lastEpochMicrosec": "1538478000000","sourceName": "some
source","startEpochMicrosec": "1538478900000","timeZoneOffset": "UTC+01:00","location": "ftpes://somelocation:22
/ftp/rop/somefile.xml.gz","compression": "gzip","fileFormatType": "org.3GPP.32.435#measCollec","
fileFormatVersion": "V9"}
```

testmapperperformance.sh

```
#!/bin/bash


rm ~/perfresults
LOOPCOUNT=$1

for i in $(seq 1 $LOOPCOUNT)
do
   START=$(date +%s)
   echo "Performance test run#$i(of $LOOPCOUNT)"

   python3 ~/delivery.py -w 10 -n 750 -r -i -H "X-DMAAP-DR-PUBLISH-ID:1" -H "X-DMAAP-DR-META:@metadata.json" -d
@APM.xml -m put http://127.0.0.1:31196/delivery/APM.xml -v

   END=$(date +%s)
   secs=$(expr $END - $START)

   echo $secs >> ~/perfresults
   if (($i < $LOOPCOUNT));then
     echo "Finished performance test run#$i(of $LOOPCOUNT), resting for a bit.."
    #sleep 13
   fi
done

echo $(awk '{s+=$1} END {print s/NR}' ~/perfresults | awk '{print int($1+0.5)}') >> ~/perftotaltime
TOTAL=$(tail -n 1 ~/perftotaltime)

TOTALTIME=$(printf '%dh:%dm:%ds\n' $(($TOTAL/3600)) $(($TOTAL%3600/60)) $(($TOTAL%60)))
echo "Average time($LOOPCOUNT runs)is: $TOTALTIME"
```

1.Deploy the simulator.

kubectl run simulator image=iamaim/simulator-perf:1.0.0 --replicas=1 --port=8888
kubectl expose deployment simulator --type=NodePort --port=8888 --target-port=8888 --name=simulator

2. Deploy pmmapper.

kubectl create configmap pmmapper-config --from-file=cert.jks.b64 --from-file=jks.pass --from- file=trust.jks.b64 --from-file=trust.pass

kubectl apply -f pmm.yaml

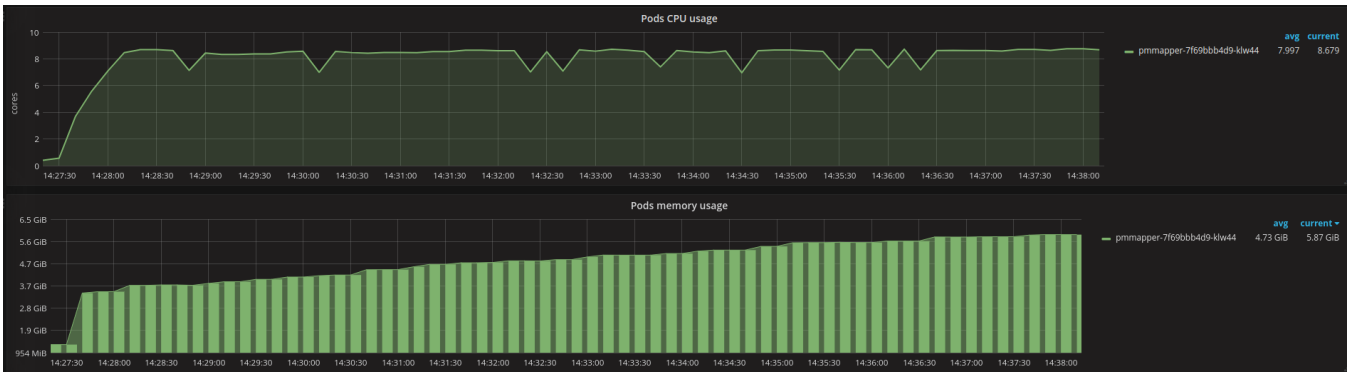
kubectl expose deployment pmmapper --type=NodePort --port=8081 --target-port=8081 --name=pmmapper

3.Run testmapperperformance.sh

```
bash testmapperperformance.sh <loop-count>
e.g.

        . testmapperperformance.sh 10
```

View your Grafana Dashboard and fine tune the view to get the cpu/memory and other useful metrics for the pmmapper pod.


Pods CPU usage
cores
10
8
6
4
2
0
14:27:30 14:28:00 14:28:30 14:29:00 14:29:30 14:30:00 14:30:30 14:31:00 14:31:30 14:32:00 14:32:30 14:33:00 14:33:30 14:34:00 14:34:30 14:35:00 14:35:30 14:36:00 14:36:30 14:37:00 14:37:30 14:38:00
avg current
pmmapper-7f69bbb4d9-klw44 7.997 8.679

Pods memory usage
6.5 GiB
5.6 GiB
4.7 GiB
3.7 GiB
2.8 GiB
1.9 GiB
954 MiB
14:27:30 14:28:00 14:28:30 14:29:00 14:29:30 14:30:00 14:30:30 14:31:00 14:31:30 14:32:00 14:32:30 14:33:00 14:33:30 14:34:00 14:34:30 14:35:00 14:35:30 14:36:00 14:36:30 14:37:00 14:37:30 14:38:00
avg current
pmmapper-7f69bbb4d9-klw44 4.73 GiB 5.87 GiB