

PM distribution to rApps

This demo package show distribution om PM (file based pm and streamed pm) from an O-DU simulator to rApps.

The main purpose of the demo is to show how a data consumer (rApp) can subscribe to data from a data producer via a job in ICS.

The demo requires a running docker and docker-compose.

Demo has been verified on Mac-OS but should work on Linux as well as windows (Git-bash)

Download the nonrtric-prototyping repo

Download repo

```
git clone "https://gerrit.nordix.org/local/oransc/nonrtric-prototyping"

cd nonrtric-prototyping/pm-data-event-demo
```

The whole package is started with a single command - ./run-all.sh. To clean all containers, use the cmd - ./down-all.sh

Start the package

```
./run-all.sh
```

(in the end of the script - hit enter to start sending fileready events from the O-DU. However, the streamed pm counters are already sending events

Start the package

```
./run-all.sh
...
...
Hit enter to start sending fileready events
...
...
```

Make sure that no errors occurs during the script execution. For example, other running container or ports already occupied by other applications may cause errors in the script.

When the script has finished, there should be 22 running containers - see example below.

NAMES	IMAGE
nonrtricgateway	nexus3.o-ran-sc.org:10004/o-ran-sc/nonrtric-gateway:1.1.0
dmaapadapterservice	nexus3.o-ran-sc.org:10004/o-ran-sc/nonrtric-dmaap-adaptor:1.0.1
controlpanel	nexus3.o-ran-sc.org:10004/o-ran-sc/nonrtric-controlpanel:2.3.0
informationservice	nexus3.o-ran-sc.org:10004/o-ran-sc/nonrtric-information-coordinator-service:1.2.1
dmaapmediatorservice	nexus3.o-ran-sc.org:10004/o-ran-sc/nonrtric-dmaap-mediator-producer:1.0.1
rapp1	python:3.8-slim-buster
rapp2	python:3.8-slim-buster
pmmapper	nexus3.onap.org:10001/onap/org.onap.dcaege2.services.pm-mapper:1.8.0
datarouter-node	nexus3.onap.org:10003/onap/dmaap/datarouter-node:2.1.10-STAGING-latest
datarouter-prov	nexus3.onap.org:10003/onap/dmaap/datarouter-prov:2.1.10-STAGING-latest
mariadb	nexus3.onap.org:10001/mariadb:10.2.14
datafile-collector	nexus3.onap.org:10001/onap/org.onap.dcaege2.collectors.datafile.datafile-app-server:1.7.1
ntsim-ng-o-du-1122	nexus3.o-ran-sc.org:10001/o-ran-sc/nts-ng-o-ran-du:1.4.3
sdnr	nexus3.onap.org:10001/onap/sdnc-image:2.2.4
ves-collector	nexus3.onap.org:10001/onap/org.onap.dcaege2.collectors.ves.vescollector:1.11.0
sdnc-web	nexus3.onap.org:10001/onap/sdnc-web-image:2.2.4
onap-dmaap	nexus3.onap.org:10001/onap/dmaap/dmaap-mr:1.1.18
kafka	nexus3.onap.org:10001/onap/dmaap/kafka111:1.0.4
zookeeper	nexus3.onap.org:10001/onap/dmaap/zookeeper:6.0.3
identity	quay.io/keycloak/keycloak:12.0.4
persistence	docker.elastic.co/elasticsearch/elasticsearch-oss:7.9.3

The O-DU simulator will continuously send events with a random counter value every 10 second. And 900 fileready events will be sent with 30 sec interval where a few counter values has counter values derived from the index of the pm file.

The flow of data can be viewed in several logs.

Events received by the VES collector

VES Collector log

```
docker exec -it ves-collector tail -f /opt/app/VESCollector/logs/input.log
```

Files fetched from the O-DU and published to the data router - by the datafile collector

Datafile collector log

```
docker logs -f datafile-collector
```

PM Mapper log

PM Mapper logs

```
docker logs -f pmmapper
```

PM files and as well as the streamed counters are sent to two simulated rAPPs, rapp1 and rapp2

Logs can be viewed (replace X with 1 or 2)

rapp logs

```
docker logs -f rappX
```

In addition, the files are also dumped to directories mounted to the simulated rapp1 and rapp2. Only the latest files is stored.

The file data.json contains the file and the event header

The file event.json contains the streamed counter and the event header

Again, replace X with 1 or 2

View received json

```
cat group6/jsondump/rappX/data.json
cat group6/jsondump/rappX/event.json
```

The set of started containers can be cleaned up at any time by this script

down-all.sh

```
./down-all.sh
```