

# Deploy DMaaP message router in nonrtric

This page is out of date.

Please see the relevant page for the latest release: e.g. [Release 1 - Run in Docker](#)

This article helps you to deploy the NON-RT-RIC components using docker compose.

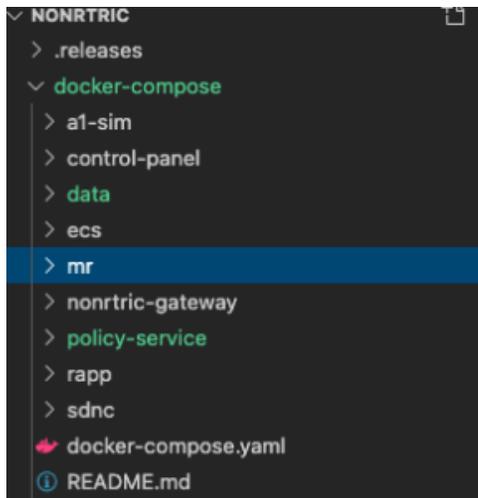
- [Prerequisite](#)
- [Installation](#)
- [NON-RT-RIC without A1 Controller, with dmaap-mr](#)
- [Init Script](#)
- [Manual test](#)
- [Message format](#)

## Prerequisite

- [Docker](#)
- [Docker Compose](#)

## Installation

Clone [nonrtric](#) and go to `nonrtric/docker-compose/`



You can see a few folders, each of them contains a docker-compose file for different components. You have the option to install the nonrtric with/without A1 controller/dmapp-mr based on your requirement.

## NON-RT-RIC without A1 Controller, with dmaap-mr

withDmaap\_nosdnc/docker-compose.yml

This file is to create nonrtric system without sdnc a1-controller, with dmapp-mr

### nosdnc with dmaap-mr

```
docker-compose -f docker-compose.yaml -f control-panel/docker-compose.yaml -f policy-service/docker-compose.yaml -f mr/docker-compose.yaml up -d
```

You can see the docker containers with `docker ps`,

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAM
ES a884db69f8c7	nexus3.onap.org:10001/onap/dmaap/dmaap-mr:1.1.18	"sh startup.sh"	12 seconds ago	Up 11 seconds	0.0.0.0:3904-3905->3904-3905/tcp	dma
ap-mr 55671d890037	nexus3.onap.org:10001/onap/dmaap/kafka111:1.0.4	"/etc/confluent/dock..."	14 seconds ago	Up 12 seconds	0.0.0.0:9092->9092/tcp, 9093/tcp	kaf
ka b9c0c93d4e19	nexus3.o-ran-sc.org:10004/o-ran-sc/a1-simulator:2.0.0	"/bin/sh -c 'src/std..."	16 seconds ago	Up 14 seconds	0.0.0.0:30003->8085/tcp, 0.0.0.0:30004->8185/tcp	a1-
sim-STD 584be418c702	nexus3.o-ran-sc.org:10004/o-ran-sc/nonrtic-controlpanel:2.0.0	"java -Xms128m -Xmx2..."	16 seconds ago	Up 13 seconds	0.0.0.0:8080->8080/tcp, 0.0.0.0:8082->8082/tcp	pol
icy-control-panel 435afdb48ad0	nexus3.o-ran-sc.org:10004/o-ran-sc/nonrtic-policy-agent:2.0.1	"java -jar /opt/app/..."	16 seconds ago	Up 13 seconds	0.0.0.0:8081->8081/tcp, 0.0.0.0:8433->8433/tcp	pol
icy-agent 1724b4eec638	nexus3.o-ran-sc.org:10004/o-ran-sc/a1-simulator:2.0.0	"/bin/sh -c 'src/std..."	16 seconds ago	Up 13 seconds	0.0.0.0:30001->8085/tcp, 0.0.0.0:30002->8185/tcp	a1-
sim-OSC dfffe1c3e04ac	nexus3.onap.org:10001/onap/dmaap/zookeeper:6.0.3	"/etc/confluent/dock..."	16 seconds ago	Up 13 seconds	2888/tcp, 0.0.0.0:2181->2181/tcp, 3888/tcp	zoo
keeper						

you should be able to see containers running as above.

## Init Script

You can create sample policy types/policy instances of both Standard & OSC Version.

Run the `fill_data.sh` script under directory `/data` to create some dummy data in the running system.

Command to load the data,

```
load data

cd nonrtic/docker-compose/data

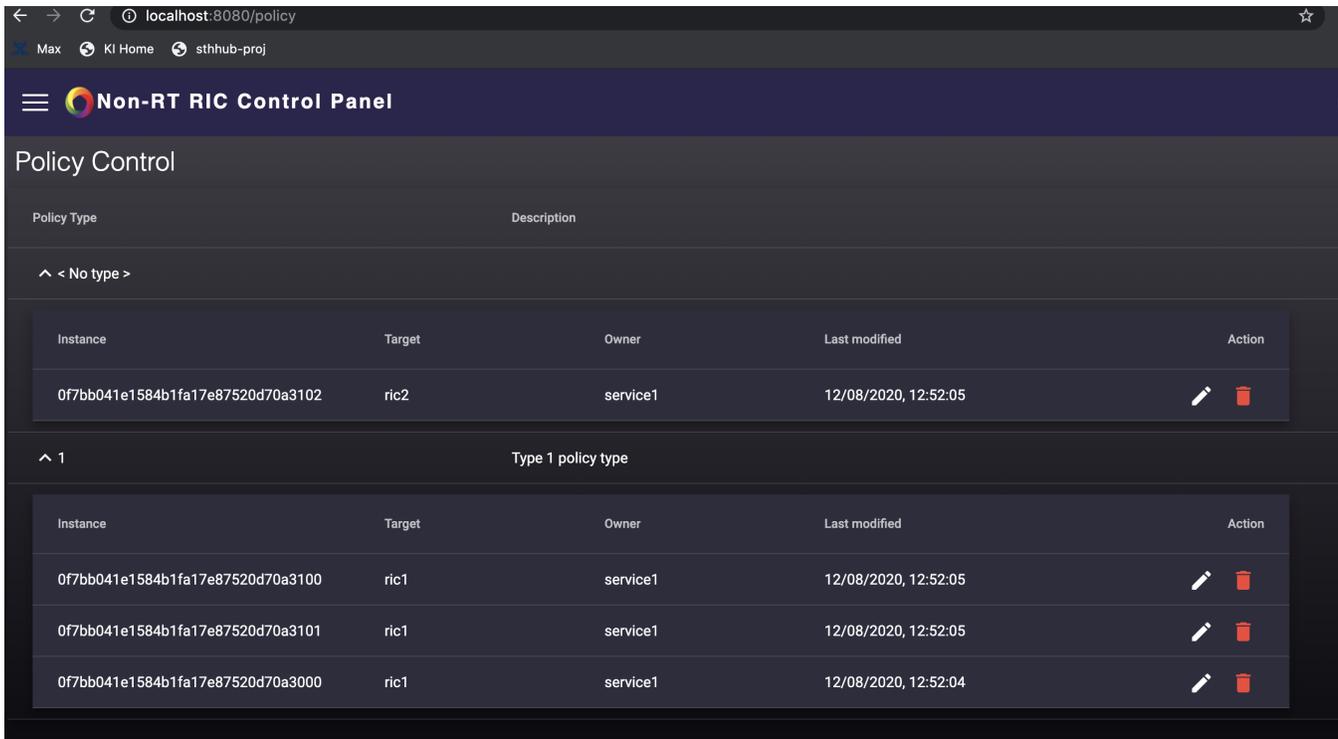
./prepareDmaapMsg.sh [dmaap-mr-port] [a1-sim-OSC-port] [a1-sim-STD-port] [http/https]
```

This will create,

- one policy type in a1-sim-OSC
- one service in policy agent
- three policies in a1-sim-OSC
- one policy in a1-sim-STD

Open the GUI to view all the policies & policy types created by the above script.

[http://localhost:\[control-panel-port\]/](http://localhost:[control-panel-port]/)



# Manual test

## send messages to dmaap-mr

```
cd nonrtric/docker-compose/data
curl -X POST "http://localhost:3904/events/A1-POLICY-AGENT-READ/" -H "accept: application/json" -H "Content-Type: application/json" --data-binary @testdata/dmaap-msg-policy-get.json
```

## pull messages from dmaap-mr

```
curl -X GET "http://localhost:3904/events/A1-POLICY-AGENT-WRITE/users/policy-agent?timeout=15000&limit=100" -H "accept: application/json" -H "Content-Type: application/json" | jq .
```

You can see the messages from dmaap-mr topic **A1-POLICY-AGENT-WRITE**, with consumer group "**user**", consumer id "**policy-agent**"

```
Chengkaiyan@Chengkais-MacBook-Pro:~/oran/docker-compose$ curl -X GET "http://localhost:3904/events/A1-POLICY-AGENT-WRITE/users/policy-agent?timeout=15000&limit=100" -H "accept: application/json" -H "Content-Type: application/json" | jq .
% Total % Received % Xferd Average Speed Time Time Time Current
         Dload Upload Total Spent Left Speed
100 2839 100 2839 0 0 4367 0 --:--:-- --:--:-- --:--:-- 4360
[
  {
    "requestId": "23343221",
    "correlationId": "1596363458549998500",
    "originatorId": "849e6c6b420",
    "type": "response",
    "message": "OK",
    "timestamp": "2020-08-02 10:17:38.550324",
    "status": "200 OK"
  },
  {
    "requestId": "23343221",
    "correlationId": "1596363458549998501",
    "originatorId": "849e6c6b421",
    "type": "response",
    "message": "OK",
    "timestamp": "2020-08-02 10:17:38.550324",
    "status": "200 OK"
  },
  {
    "requestId": "23343221",
    "correlationId": "1596363459196978900",
    "originatorId": "849e6c6b420",
    "type": "response",
    "message": "OK",
    "timestamp": "2020-08-02 10:17:39.197067",
    "status": "200 OK"
  },
  {
    "requestId": "23343221",
    "correlationId": "1596304565904621535",
    "originatorId": "849e6c6b420",
    "type": "response",
    "message": "",
    "timestamp": "2020-08-01 17:56:05.905035",
    "status": "200 OK"
  },
  {
    "requestId": "23343221",
    "correlationId": "159630456656253556",
    "originatorId": "849e6c6b420",
    "type": "response",
    "message": "",
    "timestamp": "2020-08-01 17:56:06.656949",
    "status": "200 OK"
  },
  {
    "requestId": "23343221",
    "correlationId": "159630456656253557",
    "originatorId": "849e6c6b420",
    "type": "response",
    "message": "",
    "timestamp": "2020-08-01 17:56:06.656949",
    "status": "200 OK"
  },
  {
    "requestId": "23343221",
    "correlationId": "159630456656253558",
    "originatorId": "849e6c6b420",
    "type": "response",
    "message": "",
    "timestamp": "2020-08-01 17:56:06.656949",
    "status": "200 OK"
  },
  {
    "requestId": "23343221",
    "correlationId": "1596304567017739720",
    "originatorId": "849e6c6b420",
    "type": "response",
    "message": "{\"scope\":{\"qosId\":\"qos3000\"},\"ueId\":\"ue3000\"}\",
    "timestamp": "2020-08-01 17:56:07.017744",
    "status": "200 OK"
  },
  {
    "requestId": "23343221",
    "correlationId": "1596304567017739720",
    "originatorId": "849e6c6b420",
    "type": "response",
    "message": "{\"scope\":{\"qosId\":\"qos3100\"},\"ueId\":\"ue3100\"}\",
    "timestamp": "2020-08-01 17:56:07.017744",
    "status": "200 OK"
  },
  {
    "requestId": "23343221",
    "correlationId": "1596304567017739720",
    "originatorId": "849e6c6b420",
    "type": "response",
    "message": "{\"scope\":{\"qosId\":\"qos3100\"},\"ueId\":\"ue3100\"}\",
    "timestamp": "2020-08-01 17:56:07.017744",
    "status": "200 OK"
  },
  {
    "requestId": "23343221",
    "correlationId": "1596304567017739720",
    "originatorId": "849e6c6b420",
    "type": "response",
    "message": "{\"scope\":{\"qosId\":\"qos3102\"},\"ueId\":\"ue3102\"}\",
    "timestamp": "2020-08-01 17:56:07.017744",
    "status": "200 OK"
  }
]
```

# Message format

## create a policy via dmaap-mr

```
{
  "apiVersion": "1.0",
  "operation": "PUT",
  "correlationId": "1596304565904621535",
  "originatorId": "849e6c6b420",
  "payload": {
    "scope": {
      "ueId": "ue3000",
      "qosId": "qos3000"
    },
    "qosObjective": {
      "priorityLevel": 3000
    }
  },
  "requestId": "23343221",
  "target": "policy-agent",
  "timestamp": "2020-08-01 17:56:05.905035",
  "type": "request",
  "url": "/policy?id=0f7bb041e1584b1fa17e87520d70a3000&ric=ric1&service=service1&type=1"
}
```