Deploy A1 Policy Management Service with Docker

This page is out of date.
Please see the relevant page for the latest release: e.g. Release I - Run in Docker

The A1 A1 Policy Management Service can be executed with a single docker command:

docker command with default network

```
docker run -p 8081:8081 -p 8344:8433 --name=policy-agent --mount type=bind,source=$(pwd)
/application_configuration.json,target=/opt/app/policy-agent/data/application_configuration.json nexus3.o-ran-
sc.org:10004/o-ran-sc/nonrtric-policy-agent
```

The source in the ***--mount*** command need to be correct.

If not using default network, should also add ***--network=<network-name>*** in the command.

If you want to run the container in the background, should also add ***-d*** in the command.

If you want to set network alias for PMS, add ***--network-alias=<alias-name>*** in the command.

nexus3.o-ran-sc.org:10002/o-ran-sc/nonrtric-policy-agent:2.0.0 is the Bronze release version.

nexus3.o-ran-sc.org:10004/o-ran-sc/nonrtric-policy-agent is the latest version from master branch.

Example with all above arguments:

docker command with user defined network

```
docker run -d -p 8081:8081 -p 8344:8433 --name policy-agent --network=nonrtric-docker-net --network-alias
policy-agent-container --mount type=bind,source=$(pwd)/application_configuration.json,target=/opt/app/policy-
agent/data/application_configuration.json nexus3.o-ran-sc.org:10004/o-ran-sc/nonrtric-policy-agent
```

To execute the Policy Management Service you will need an “application_configuration.json” file to configure the service.
This configuration file is then mounted into the correct place in the docker container using the *--mount* parameter above (*--mount type=bind,source=$(pwd)/application_configuration.json,target=/opt/app/policy-agent/data/application_configuration.json*)

- For more information on configuration options see page: Release B - Build/Run
- NOTE: There should not be any comma in the end of last configuration.

application_configuration.json

```
{
    "config":{
       "ric":[
          {
             "name":"ric1",
             "baseUrl":"http://a1-sim-OSC:8085/",
             "managedElementIds":[
                "kista_1",
                "kista_2"
             ]
          },
          {
             "name":"ric2",
             "baseUrl":"http://a1-sim-STD:8085/",
             "managedElementIds":[
                "kista_1",
                "kista_2"
             ]
          }
       ],
       "streams_publishes":{
          "dmaap_publisher":{
             "type":"message_router",
             "dmaap_info":{
                "topic_url":"http://dmaap-mr:3904/events/A1-POLICY-AGENT-WRITE"
             }
          }
       },
       "streams_subscribes":{
          "dmaap_subscriber":{
             "type":"message_router",
             "dmaap_info":{
                "topic_url":"http://dmaap-mr:3904/events/A1-POLICY-AGENT-READ/users/policy-agent?
timeout=15000&limit=100"
             }
          }
       }
    }
 }
```

Some sample CURL commands :

- For more information about the A1 Policy Management Service REST APIs see https://docs.o-ran-sc.org/projects/o-ran-sc-nonrtric/en/latest/policy-agent-api.html

1. Register a sample service "*service1*" in the Policy Management Service:

Curl: Create a sample service

```
curl -k -X PUT -skw %{http_code} -H accept:application/json -H Content-Type:application/json http://localhost:
8081/service --data '{"callbackUrl": "http://callback-receiver:8090/callbacks/1","keepAliveIntervalSeconds":
"3600","serviceName": "service1"}'
```

2. Create a new sample A1 Policy "*2001*" of type "*1*" for the service "*service1*" in one of the near-RT-RICs ("*ric1*") configured in the config file above:

(This assumes that the near-RT-RIC "*ric1*" uses a version of the A1 protocol that supports Policy Types (e.g. OSC-A1-v2.1.0), and the near-RT-RIC supports Policy Type "*1*", and the data passed for the Policy object complies with the schema for that Policy Type.)

Curl: Create a sample A1 Policy

```
       curl -k -X PUT -skw %{http_code} -H accept:application/json -H Content-Type:application/json
"http://localhost:8081/policy?id=2001&ric=ric1&service=service1&type=1" --data '{"scope": {"ueId": "ue3100","
qosId": "qos3100"},"qosObjectives": {"priorityLevel": 3101}}'
```

3. Create a new sample A1 Policy "*3100*" of *no* type for the service "*service1*" in the other near-RT-RICs ("*ric2"*) configured in the config file above:

(This assumes that the near-RT-RIC "*ric2*" uses a version of the A1 protocol that does not support Policy Types (e.g. A1-AP-v1.1) but the data passed for the Policy object can be interpreted by the near-RT-RIC)

Curl: Create a sample A1 Policy

```
curl -k -X PUT -skw %{http_code} -H accept:application/json -H Content-Type:application/json "http://localhost:
8081/policy?id=3100&ric=ric2&service=service1" --data '{"scope": {"ueId": "ue3100","qosId": "qos3100"},"
qosObjectives": {"priorityLevel": 3100}}'
```

4. Get status for the A1 Policy Management Service:

Curl: Get status information for the A1 Policy Management Service

```
curl -skw %{http_code} http://localhost:8081/status
```