

# Release A - Build/Run

This page describes how to get the release A version of NonRTRIC up and running locally with two separate NearRT-RIC simulator docker containers.

All four components of the NonRTRIC repo run inside docker containers and communicate via a docker network. Details of the architecture can be found from [Release A](#) page.

- [Project Requirements](#)
- [Build Code](#)
  - [Configure policy-agent](#)
  - [Verify dashboard configuration](#)
  - [Build the code and create docker images](#)
- [Run A1 Controller Docker Container](#)
- [Run Policy-agent Docker Container](#)
  - [Note on automatic A1 Client selection](#)
- [Run Non-RT-RIC Dashboard Docker Container](#)
- [Open Daylight GUI](#)
- [Dashboard UI](#)

## Project Requirements

- Java 11 (make sure that JAVA\_HOME environment variable points to correct Java version)
- Maven 3.6
- Docker and docker-compose (latest)

## Build Code

Download the nonrtric repo:

```
git clone "https://gerrit.o-ran-sc.org/r/nonrtric" -b Amber
```

## Configure policy-agent

To support local test with two separate NearRT-RIC simulator instances the following modifications must be done:

- change the '**a1ControllerBaseUrl**' in **nonrtric/policy-agent/config/application.yaml** file of policy-agent to <http://a1-controller-container:8181>
  - The above hostname and port values come from and must match those defined in **nonrtric/sdnc-a1-controller/oam/installation/src/main/yaml/docker-compose.yml**
  - The default URL to be replaced (<http://sdnc.onap:8282>) is used in Kubernetes-based ONAP deployment
- The following step is required only if you want to run more than the default ric1 nearRT-RIC simulator instance
  - replace the content of **nonrtric/policy-agent/config/application\_configuration.json** with the configuration below
  - any defined ric names must match the given docker container names in nearRT-RIC simulator startup - port is always the simulator's internal 8085

#### application\_configuration.yaml

```
{
  "config": {
    "//description": "Application configuration",
    "ric": [
      {
        "name": "ric1",
        "baseUrl": "http://ric1:8085/",
        "managedElementIds": [
          "kista_1",
          "kista_2"
        ]
      },
      {
        "name": "ric2",
        "baseUrl": "http://ric2:8085/",
        "managedElementIds": [
          "kista_3",
          "kista_4"
        ]
      }
    ]
  }
}
```

## Verify dashboard configuration

⚠ Note that `nonrtic/dashboard/webapp-backend/src/main/resources/application.properties` default property value `policycontroller.url.prefix = http://policy-agent-container:8081` assumes that policy agent is started with name `policy-agent-container` and exposes port 8081 (as is done below)!

## Build the code and create docker images

To build docker images of `sdnc-a1-controller`, `policy-agent`, `near-rt-ric-simulator` and `nonrtic-dashboard`:

```
cd nonrtic
mvn clean install -Dmaven.test.skip=true
```

## Run A1 Controller Docker Container

- A1 Controller must be started first to set up docker network
- Change directory to:  
`cd nonrtic/sdnc-a1-controller/oam/installation/src/main/yaml`
- Run docker container using the command below  
`docker-compose up -d a1-controller`
  - the container is ready once message "Healthcheck Passed in XX seconds." appears
  - note that certificate-related errors "cp: can't stat '/opt/opendaylight/current/certs/': No such file or directory" and "Error: File not found in path entered" can be ignored
- The Karaf logs of A1 controller can be seen using the following commands:  
`docker exec -it a1-controller-container sh`  
`cd /opt/opendaylight/data/log/`  
`tail -f karaf.log`

## Run Near-RT-RIC Simulator Docker Containers

- Start docker containers for each ric defined in `nonrtic/policy-agent/config/application_configuration.json` in previous steps using following commands (in this example for `ric1` and `ric2`):  
`docker run -p 8085:8085 --network=nonrtic-docker-net --name=ric1 o-ran-sc/near-rt-ric-simulator:1.0.1-SNAPSHOT`  
`docker run -p 8086:8085 --network=nonrtic-docker-net --name=ric2 o-ran-sc/near-rt-ric-simulator:1.0.1-SNAPSHOT`
- Change directory to:

```
cd nonrttric/near-rt-ric-simulator/ric-plt/a1
```

- Put an example policy\_type into all the started near-rt-ric-simulator instances by running these curl commands (in this example to ric1 exposed to port 8085 and ric2 exposed to port 8086):  

```
curl -X PUT -v "http://localhost:8085/policytypes/STD_QoSNUdging_0.2.0" -H "accept: application/json" \
-H "Content-Type: application/json" --data-binary @policy_type_STD_QoSNUdging_0.2.0.json
```

  

```
curl -X PUT -v "http://localhost:8086/policytypes/STD_QoSNUdging_0.2.0" -H "accept: application/json" \
-H "Content-Type: application/json" --data-binary @policy_type_STD_QoSNUdging_0.2.0.json
```

## Run Policy-agent Docker Container

- Run docker container using this command once A1 Controller and simulators have been fully started (see note below)::  

```
docker run -p 8081:8081 --network=nonrttric-docker-net --name=policy-agent-container o-ran-sc/nonrttric-policy-agent:1.0.0-SNAPSHOT
```

### ⚠ Note on automatic A1 Client selection

Once policy-agent is up and running, it establishes connections to all configured NearRT-RICs. It has multiple A1 clients to select from and selection is done dynamically based on connectivity. Selection of A1 client is not clearly visible for user, but policy-agent logs can be grepped to see which client was selected.

The two grepping examples below show how either SDNC\_OSC A1 client or STD A1 client was selected - these translate into options A and B described on the [Testing End to End call in release A](#) page.

#### SDNC A1 Client

```
$ docker logs policy-agent | grep "protocol version"
2020-03-30 05:33:21.765 DEBUG 1 --- [or-http-epoll-2] o.o.policyagent.clients.A1ClientFactory : Recover ric:
ric2, protocol version:SDNC_OSC
2020-03-30 05:33:21.784 DEBUG 1 --- [or-http-epoll-1] o.o.policyagent.clients.A1ClientFactory : Recover ric:
ric1, protocol version:SDNC_OSC
```

#### STD A1 Client

```
$ docker logs policy-agent | grep "protocol version"
2020-04-03 12:06:54.548 DEBUG 1 --- [or-http-epoll-1] o.o.policyagent.clients.A1ClientFactory : Recover ric:
ric1, protocol version:STD_V1
2020-04-03 12:06:54.552 DEBUG 1 --- [or-http-epoll-2] o.o.policyagent.clients.A1ClientFactory : Recover ric:
ric2, protocol version:STD_V1
```

## Run Non-RT-RIC Dashboard Docker Container

- Run docker container using this command:  

```
docker run -p 8080:8080 --network=nonrttric-docker-net o-ran-sc/nonrttric-dashboard:1.0.1-SNAPSHOT
```

## Open Daylight GUI

- Open Daylight GUI can be accessed by pointing web-browser to this URL:  
<http://localhost:8282/apidoc/explorer/index.html>  
Username/password: admin/Kp8bJ4SXszM0WXIhak3eHlcse2gAw84vaoGGmJvUy2U

## Dashboard UI

- Dashboard UI can be accessed by pointing the web-browser to this URL:  
<http://localhost:8080/>