Deploy callback_reciever with docker commands

This page is out of date.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

callback_reciever repo:

https://gerrit.o-ran-sc.org/r/gitweb?p=nonrtric.git;a=tree;f=test/cr;hb=refs/changes/02/4502/5

callback receiver - a stub interface to receive callbacks

The callback receiver is intended for function tests to simulate a RAPP.
The callback receiver exposes the read and write urls, used by the agent, as configured in service.

The callback receiver receives notifications from PMS when synchronisation happens between PMS and RICs.

Ports and certificates

The CR normally opens the port 8090 for http. If a certificate and a key are provided the simulator will also open port 8091 for https.
The certificate and key shall be placed in the same dir and the dir shall be mounted to /usr/src/app/cert in the container.

Port	Protocol
8090	http
8091	https

The dir cert contains a self-signed cert. Use the script generate_cert_and_key.sh to generate a new certificate and key. The password of the certificate must be set 'test'.
The same urls are available on both the http port 8090 and the https port 8091. If using curl and https, the flag -k shall be given to make curl ignore checking the certificate.

Control interface

The control interface can be used by any test script.
The following REST operations are available:

Send a message to CR:
URI and payload, (PUT or POST): /callbacks/<id> <json array of response messages>
response: OK 200 or 500 for other errors

Metrics - counters
There are a number of counters that can be read to monitor the message processing. Do a http GET on any of the current counters and an integer value will be returned with http response code 200.
/counter/received_callbacks - The total number of received callbacks
/counter/fetched_callbacks - The total number of fetched callbacks
/counter/current_messages - The current number of callback messages waiting to be fetched

Build and start

Build image:

Build image

```
docker build -t callback-receiver .
```

Start the image on both http and https:

Start the image on both http and https:

```
docker run -it -p 8090:8090 -p 8091:8091 callback-receiver
```

It will listen to http 8090 port and https 8091 port(using default certificates) at the same time.

By default, this image has default certificates under ***/usr/src/app/cert***
file "***cert.crt***" is the certificate file
file "***key.crt***" is the key file
file "***generate_cert_and_key.sh***" is a shell script to generate certificate and key
file "***pass***" stores the password when you run the shell script

This certificates/key can be overriden by mounting a volume when using "docker run" or "docker-compose"
In '***docker run***', use field:

Mount external cert/key

```
docker run -it -p 8090:8090 -p 8091:8091 -v "/PATH_TO_CERT/cert:/usr/src/app/cert" callback-receiver
```

In 'docker-compose.yml', use field:
volumes:
- ./certificate:/usr/src/app/cert:ro

The script ***crstub-build-start.sh*** do the above two steps in one go. This starts the callback-receiver container in stand-alone mode for basic test.

If the callback-receiver should be executed manually with the agent, replace docker run with this command to connect to the docker network with the correct service name (--name shall be aligned with the other components, i.e. the host named given in all callback urls).

start callback_reciever with network parameter

```
docker run -it -p 8090:8090 -p 8091:8091 --network nonrtric-docker-net --name callback-receiver callback-receiver
```

Start the image on http only:

start the image on http only

```
docker run -it -p 8090:8090 callback-receiver
```

Basic test

Basic test is made with the script:

run basic test with script

```
basic_test.sh nonsecure|secure
```

which tests all the available urls with a subset of the possible operations.

Use the script:

start the container with script

```
cr-build-start.sh
```

to start the callback-receiver in a container first.