

Release I - Run in Kubernetes

This wiki describes how to deploy the NONRTRIC components within Kubernetes cluster.

- [NONRTRIC Architecture](#)
- [Prerequisites](#)
- [Preparations](#)
- [Configuration of components to install](#)
- [Installation](#)
- [RANPM ONLY Installation](#)
 - [Prerequisites](#)
- [Requirement on kubernetes](#)
- [Other requirements](#)
 - [Before installation](#)
 - [Installation\(RANPM Only\)](#)
 - [Uninstallation\(RANPM Only\)](#)
 - [Exposed ports to APIs](#)
 - [Keycloak API](#)
 - [OPA rules bundle server](#)
 - [Information coordinator Service](#)
 - [Ves-Collector](#)
 - [Exposed ports to admin tools](#)
 - [Keycloak admin console](#)
 - [Redpanda console](#)
 - [Minio web](#)
 - [Influx db](#)
- [Result of the installation](#)
- [Troubleshooting](#)
- [Un-installation](#)
- [Introduction to Helm Charts](#)

NONRTRIC Architecture

NONRTRIC comprises several components,

1. Control Panel
2. Policy Management Service
3. Information Coordinator Service
4. Non RT RIC Gateway (reuse of existing kong proxy is also possible)
5. R-App catalogue Service
6. Enhanced R-App catalogue Service
7. A1 Simulator (3 A1 interface versions - previously called Near-RT RIC A1 Interface)
8. A1 Controller (currently using SDNC from ONAP)
9. Helm Manager
10. Dmaap Adapter Service
11. Dmaap Mediator Service
12. Use Case rApp O-DU Slice Assurance
13. Use Case rAPP O-RU Closed loop recovery
14. CAPIF core
15. RANPM
16. RAPP Manager
17. DME Participant

In the [IT/Dep](#) repo, there are helm charts for each these components. In addition, there is a chart called nontric, which is a composition of the components above.

Prerequisites

- `kubernetes v1.19 +`
- `docker` and `docker-compose` (latest)
- `git`
- Text editor, e.g. `vi`, `notepad`, `nano`, etc.
- `helm`
- `helm3`
- ChartMuseum to store the HELM charts on the server, multiple options are available:
 - Execute the install script:

```
./dep/smo-install/scripts/layer-0/0-setup-charts-museum.sh
```
 - Install chartmuseum manually on port 18080 (<https://chartmuseum.com/#Instructions>, <https://github.com/helm/chartmuseum>)

Preparations

Download the the [it/dep](#) repository. Assuming the contents to be pulled from [i-release branch](#). If needed, may switch to [master](#) branch.

Clone repo

```
git clone "https://gerrit.o-ran-sc.org/r/it/dep" -b i-release  
  
or of the branch is not yet created:  
  
git clone "https://gerrit.o-ran-sc.org/r/it/dep"
```

Configuration of components to install

It is possible to configure which of nonrtic components to install, including the controller and a1 simulators. This configuration is made in the override for the helm package. Edit the following file

Edit override file

```
<editor> dep/RECIPE_EXAMPLE/NONRTIC/example_recipe.yaml
```

The file shown below is a snippet from the override `example_recipe.yaml`.

All parameters beginning with 'install' can be configured 'true' for enabling installation and 'false' for disabling installation.

For the parameters `installNonrticgateway` and `installKong`, only one can be enabled.

There are many other parameters in the file that may require adaptation to fit a certain environment. For example `hostname`, `namespace` and `port` to message router etc. These integration details are not covered in this guide.

Edit the override file

```
nonrtric:
  installPms: true
  installAlcontroller: true
  installAlsimulator: true
  installControlpanel: true
  installInformationservice: true
  installRappcatalogueservice: true
  installRappcatalogueEnhancedservice: true
  installNonrtricgateway: true
  installKong: false
  installDmaapadapterservice: true
  installDmaapmediatorservice: true
  installHelmmmanager: true
  installOruclosedlooprecovery: true
  installOdualiceassurance: true
  installCapifcore: true
  installRanpm: true
  installrappmanager: true
  installdmeparticipant: true

  volume1:
    # Set the size to 0 if you do not need the volume (if you are using Dynamic Volume Provisioning)
    size: 2Gi
    storageClassName: pms-storage
  volume2:
    # Set the size to 0 if you do not need the volume (if you are using Dynamic Volume Provisioning)
    size: 2Gi
    storageClassName: ics-storage
  volume3:
    size: 1Gi
    storageClassName: helmmmanager-storage

...
...
...
```

Installation

There is a script that packs and installs the components by using the `helm` command. The installation uses a values override file like the one shown above. This example can be run like this:

Deploy Nonrtric

```
sudo dep/bin/deploy-nonrtric -f dep/nonrtric/RECIPE_EXAMPLE/example_recipe.yaml
```

RANPM ONLY Installation

Prerequisites

The ranpm setup works on linux/MacOS or on windows via WSL using a local or remote kubernetes cluster.

- local kubectl
- kubernetes cluster
- local docker for building images

It is recommended to run the ranpm on a kubernetes cluster instead of local docker-desktop etc as the setup requires a fair amount of computer resources.

Requirement on kubernetes

The demo set can be run on local or remote kubernetes. Kubectl must be configured to point to the applicable kubernetes instance. Nodeports exposed by the kubernetes instance must be accessible by the local machine - basically the kubernetes control plane IP needs to be accessible from the local machine.

- Latest version of istio installed

Other requirements

- helm3
- bash
- cmd 'envsubst' must be installed (check by cmd: 'type envsubst')
- cmd 'jq' must be installed (check by cmd: 'type jq')
- keytool
- openssl

Before installation

The following images need to be built manually. If remote or multi node cluster is used, then an image repo needs to be available to push the built images to. If external repo is used, use the same repo for all built images and configure the reponame in [helm/global-values.yaml](#)

The parameter value of extimagerepo shall have a trailing /.

Build the following images (Build instruction in each directory)

- https-server
- pm-rapp

Installation(RANPM Only)

The installation is made by a few scripts. The main part of the ranpm is installed by a single script. Then, additional parts can be added on top. All installations in kubernetes is made by helm charts.

The following scripts are provided for installing (install-nrt.sh must be installed first):

The kubeconfig file of the local cluster should be aligned to the cluster's control plane node's internal IP

- install-nrt.sh : Installs the main parts of the ranpm setup
- install-pm-log.sh : Installs the producer for influx db
- install-pm-influx-job.sh : Sets up an alternative job to produce data stored in influx db.
- install-pm-rapp.sh : Installs a rapp that subscribe and print out received data

Get Pods

```
>sudo kubectl get po -n nonrtric
```

NAME	READY	STATUS	RESTARTS	AGE
bundle-server-7f5c4965c7-vsgn7	1/1	Running	0	8m16s
dfc-0	2/2	Running	0	6m31s
influxdb2-0	1/1	Running	0	8m15s
informationservice-776f789967-dxqrj	1/1	Running	0	6m32s
kafka-1-entity-operator-fcb6f94dc-fkx8z	3/3	Running	0	7m17s
kafka-1-kafka-0	1/1	Running	0	7m43s
kafka-1-zookeeper-0	1/1	Running	0	8m7s
kafka-client	1/1	Running	0	10m
kafka-producer-pm-json2influx-0	1/1	Running	0	6m32s
kafka-producer-pm-json2kafka-0	1/1	Running	0	6m32s
kafka-producer-pm-xml2json-0	1/1	Running	0	6m32s
keycloak-597d95bbc5-nsqww	1/1	Running	0	10m
keycloak-proxy-57f6c97984-hl2b6	1/1	Running	0	10m
message-router-7d977b5554-8tp5k	1/1	Running	0	8m15s
minio-0	1/1	Running	0	8m15s
minio-client	1/1	Running	0	8m16s
opa-ics-54fdf87d89-jt5rs	1/1	Running	0	6m32s
opa-kafka-6665d545c5-ct7dx	1/1	Running	0	8m16s
opa-minio-5d6f5d89dc-xls9s	1/1	Running	0	8m16s
pm-producer-json2kafka-0	2/2	Running	0	6m32s
pm-rapp	1/1	Running	0	67s
pmlog-0	2/2	Running	0	82s
redpanda-console-b85489cc9-nqqpm	1/1	Running	0	8m15s
strimzi-cluster-operator-57c7999494-kvk69	1/1	Running	0	8m53s
ves-collector-bd756b64c-wz28h	1/1	Running	0	8m16s
zoo-entrance-85878c564d-59gp2	1/1	Running	0	8m16s

Get Pods

```
>sudo kubectl get po -n ran
```

NAME	READY	STATUS	RESTARTS	AGE
pm-https-server-0	1/1	Running	0	32m
pm-https-server-1	1/1	Running	0	32m
pm-https-server-2	1/1	Running	0	32m
pm-https-server-3	1/1	Running	0	32m
pm-https-server-4	1/1	Running	0	31m
pm-https-server-5	1/1	Running	0	31m
pm-https-server-6	1/1	Running	0	31m
pm-https-server-7	1/1	Running	0	31m
pm-https-server-8	1/1	Running	0	31m
pm-https-server-9	1/1	Running	0	31m

Uninstallation(RANPM Only)

There is a corresponding uninstall script for each install script. However, it is enough to just run `uninstall-nrt.sh` and `uninstall-pm-rapp.sh`.

Exposed ports to APIs

All exposed APIs on individual port numbers (nodeporta) on the address of the kubernetes control plane.

Keycloak API

Keycloak API accessed via proxy (proxy is needed to make keycloak issue token with the internal address of keycloak).

- nodeport: 31784

OPA rules bundle server

Server for posting updated OPA rules.

- nodeport: 32201

Information coordinator Service

Direct access to ICS API. -nodeports (http and https): 31823, 31824

Ves-Collector

Direct access to the Ves-Collector

- nodeports (http and https): 31760, 31761

Exposed ports to admin tools

As part of the ranpm installation, a number of admin tools are installed. The tools are accessed via a browser on individual port numbers (nodeports) on the address of the kubernetes control plane.

Keycloak admin console

Admin tool for keycloak.

- nodeport : 31788
- user: admin
- password: admin

Redpanda consule

With this tool the topics, consumer etc can be viewed.

- nodeport: 31767

Minio web

Browser for minio filestore.

- nodeport: 31768
- user: admin
- password: adminadmin

Influx db

Browser for influx db.

- nodeport: 31812
- user: admin
- password: mySuP3rS3cr3tT0keN

Result of the installation

The installation will create one helm release and all created kubernetes objects will be put in a namespace. This name is 'nonrtic' and cannot be changed.

Once the installation is done you can check the created kubernetes objects by using command `kubectl`.

Example : Deployed pods when all components are enabled:

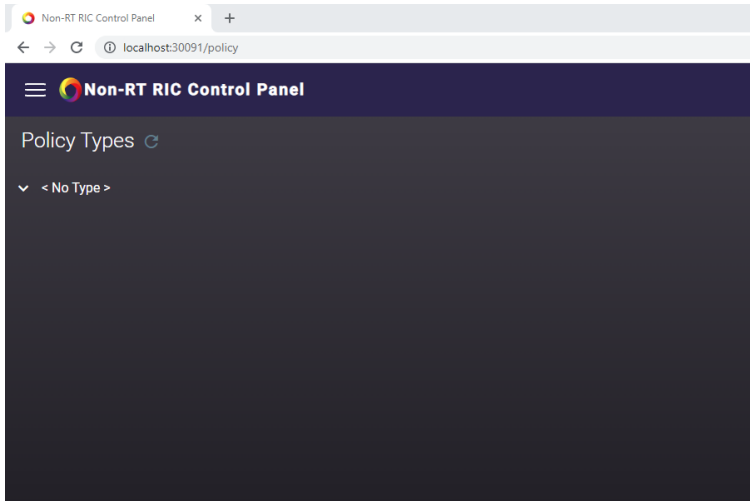
Get Pods

```
>sudo kubectl get po -A
```

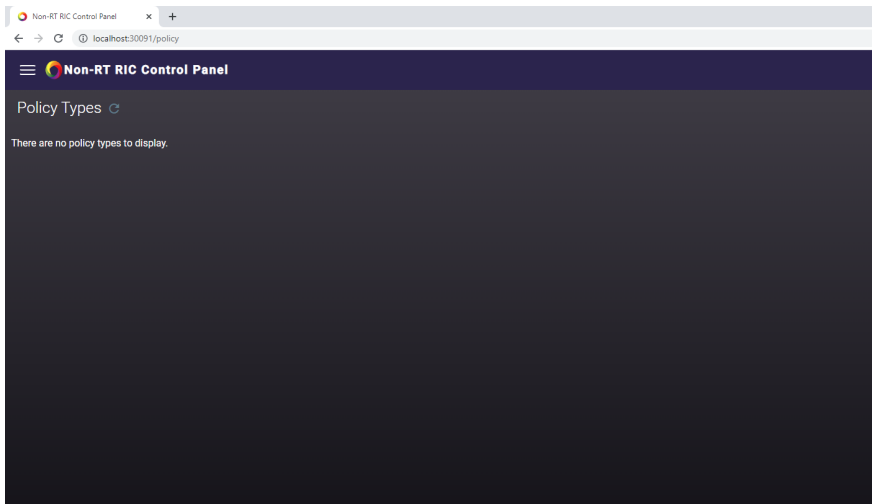
NAME	READY	STATUS	RESTARTS	AGE
al-sim-osc-0	1/1	Running	0	2m27s
al-sim-osc-1	1/1	Running	0	117s
al-sim-std-0	1/1	Running	0	2m27s
al-sim-std-1	1/1	Running	0	117s
al-sim-std2-0	1/1	Running	0	2m27s
al-sim-std2-1	1/1	Running	0	117s
alcontroller-558776cc7b-8rhdd	1/1	Running	0	2m27s
capifcore-684b458c9b-w297x	1/1	Running	0	2m27s
controlpanel-889b5dfbf-b8tgd	1/1	Running	0	2m27s
db-75c5789d97-nvjtw	1/1	Running	0	2m27s
dmaapadapterservice-0	1/1	Running	0	2m27s
dmaapmediatorservice-0	1/1	Running	0	2m27s
helmmanger-0	1/1	Running	0	2m27s
informationservice-0	1/1	Running	0	2m27s
nonrtrcgateway-7b7d485dd4-j8hnz	1/1	Running	0	2m27s
orufhrecovery-6d97d6ccf-ghknd	1/1	Running	0	2m27s
policymanagementservice-0	1/1	Running	0	2m27s
ransliceassurance-7d788d7556-95trk	1/1	Running	0	2m27s
rappcatalogueenhancedservice-764c47f7fb-s75hf	1/1	Running	0	2m27s
rappcatalogueservice-66c7bf7d98-2ldjc	1/1	Running	0	2m27s
rappmanager-0	1/1	Running	0	2m27s
dmeparticipant-84497bcbd4-jr522	1/1	Running	0	2m27s
bundle-server-7f5c4965c7-vsgn7	1/1	Running	0	8m16s
dfc-0	2/2	Running	0	6m31s
influxdb2-0	1/1	Running	0	8m15s
informationservice-776f789967-dxqrj	1/1	Running	0	6m32s
kafka-1-entity-operator-fcb6f94dc-fkx8z	3/3	Running	0	7m17s
kafka-1-kafka-0	1/1	Running	0	7m43s
kafka-1-zookeeper-0	1/1	Running	0	8m7s
kafka-client	1/1	Running	0	10m
kafka-producer-pm-json2influx-0	1/1	Running	0	6m32s
kafka-producer-pm-json2kafka-0	1/1	Running	0	6m32s
kafka-producer-pm-xml2json-0	1/1	Running	0	6m32s
keycloak-597d95bbc5-nsgww	1/1	Running	0	10m
keycloak-proxy-57f6c97984-hl2b6	1/1	Running	0	10m
message-router-7d977b5554-8tp5k	1/1	Running	0	8m15s
minio-0	1/1	Running	0	8m15s
minio-client	1/1	Running	0	8m16s
opa-ics-54fdf87d89-jt5rs	1/1	Running	0	6m32s
opa-kafka-6665d545c5-ct7dx	1/1	Running	0	8m16s
opa-minio-5d6f5d89dc-xls9s	1/1	Running	0	8m16s
pm-producer-json2kafka-0	2/2	Running	0	6m32s
pm-rapp	1/1	Running	0	67s
pmlog-0	2/2	Running	0	82s
redpanda-console-b85489cc9-nqqpm	1/1	Running	0	8m15s
strimzi-cluster-operator-57c7999494-kvk69	1/1	Running	0	8m53s
ves-collector-bd756b64c-wz28h	1/1	Running	0	8m16s
zoo-entrance-85878c564d-59gp2	1/1	Running	0	8m16s
pm-https-server-0	1/1	Running	0	32m
pm-https-server-1	1/1	Running	0	32m
pm-https-server-2	1/1	Running	0	32m
pm-https-server-3	1/1	Running	0	32m
pm-https-server-4	1/1	Running	0	31m
pm-https-server-5	1/1	Running	0	31m
pm-https-server-6	1/1	Running	0	31m
pm-https-server-7	1/1	Running	0	31m
pm-https-server-8	1/1	Running	0	31m
pm-https-server-9	1/1	Running	0	31m

Troubleshooting

- After successful installation, control panel shows "No Type" as policy type as shown below.



- If there is no policy type shown and UI looks like below, then the setup can be investigated with below steps (It could be due to synchronization delay as well, It gets fixed automatically after few minutes)



- Verify the A1 PMS logs to make sure that the connection between A1 PMS and a1controller is successful.
 - Command to check pms logs

Kubernetes command to get PMS logs

```
kubectl logs policymanagementservice-0 -n nonrttric
```

- Command to enable debug logs in PMS (Command below should be executed inside k8s pods or the host address needs to be updated with the relevant port forwarding)

Enabling debug logs in PMS

```
curl --request POST \  
--url http://policymanagementservice:9080/actuator/loggers/org.onap.ccsdk.oran.alpolicymanagementservice \  
--header 'Content-Type: application/json' \  
--data '{ "configuredLevel": "DEBUG" }'
```

- Try removing the controller information in specific simulator configuration and verify the simulator are working without a1controller.

application_configuration.json

```
...
{
  "name": "ric1",
  "baseUrl": "https://a1-sim-osc-0.a1-sim:8185",
  "managedElementIds": [
    "kista_1",
    "kista_2"
  ]
},
...
```

- `curl` can be used in control panel pod.

Un-installation

There is a script that uninstalls the NONRTRIC components. It is simply run like this:

Undeploy Nonrtric

```
sudo dep/bin/undeploy-nonrtric
```

Introduction to Helm Charts

In NONRTRIC we use Helm chart as a packaging manager for kubernetes. Helm chart helps developer to package, configure & deploy the application and services into kubernetes environment.

For more information you could refer to below links,

<https://helm.sh/docs/intro/quickstart/>