

# Release F - Run in Kubernetes

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

- [NONRTRIC Architecture](#)
- [Preparations](#)
- [Configuration of components to install](#)
- [Installation](#)
- [Result of the installation](#)
- [Troubleshooting](#)
- [Un-installation](#)
- [Introduction to Helm Chart](#)

## 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. A1 Simulator (3 A1 interface versions - previously called Near-RT RIC A1 Interface)
7. A1 Controller (currently using SDNC from ONAP)
8. Helm Manager
9. Dmaap Adapter Service
10. Dmaap Mediator Service
11. Use Case rApp O-DU Slice Assurance
12. Use Case rAPP O-RU Closed loop recovery

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

## Preparations

Download the the it/dep repository. At time of writing there is no branch for f-release so cloning can be made from master. Pls check if the branch exist before cloning from master.

### Clone repo

```
git clone "https://gerrit.o-ran-sc.org/r/it/dep" -b f-release  
  
or  
  
git clone "https://gerrit.o-ran-sc.org/r/it/dep"
```

## Configuration of components to install

It is possible to configure which of nonrtric 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/NONRTRIC/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 installNonrtricgateway 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.

#### Editor override file

```
nonrtric:
  installPms: true
  installAlcontroller: true
  installAlsimulator: true
  installControlpanel: true
  installInformationservice: true
  installRappcatalogueservice: true
  installNonrtricgateway: true
  installKong: false
  installDmaapadapterservice: true
  installDmaapmediatorservice: true
  installHelmmmanager: true
  installOruclosedlooprecovery: true
  installOdualiceassurance: 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
```

## 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 'nonrtric' 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:

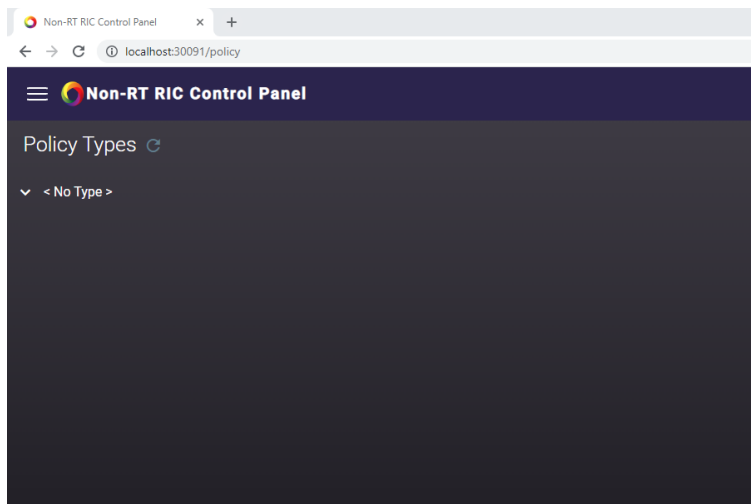
## Get Pods

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

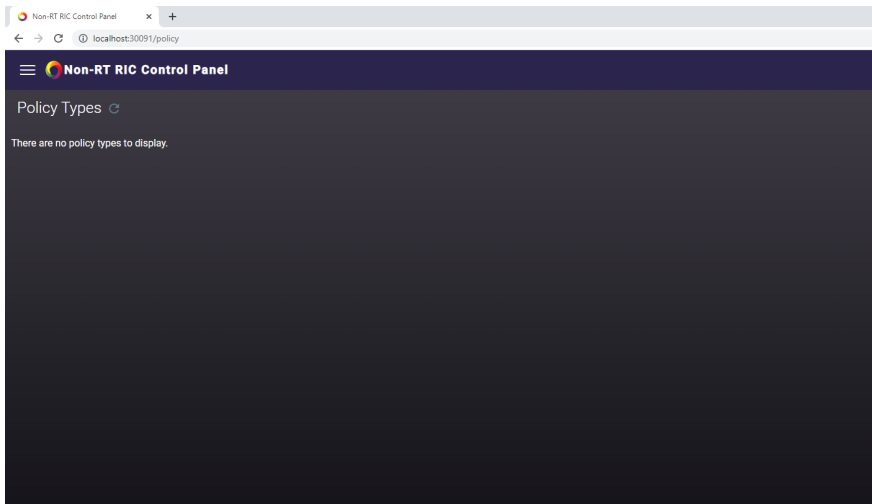
NAME	READY	STATUS	RESTARTS	AGE
al-sim-osc-0	1/1	Running	0	12m
al-sim-osc-1	1/1	Running	0	10m
al-sim-std-0	1/1	Running	0	12m
al-sim-std-1	1/1	Running	0	10m
al-sim-std2-0	1/1	Running	0	12m
al-sim-std2-1	1/1	Running	0	3m57s
alcontroller-64c5b7fc56-hjx6l	1/1	Running	0	12m
controlpanel-6bf7c4bf79-m6hgl	1/1	Running	0	12m
db-76d79cd769-mngm4	1/1	Running	0	12m
dmaapadapterservice-0	1/1	Running	0	12m
dmaapmediatorservice-0	1/1	Running	0	12m
helmmanager-0	1/1	Running	0	12m
informationservice-0	1/1	Running	0	12m
nonrtricgateway-677988d5c7-rfm88	1/1	Running	0	12m
odusliceassurance-cd5b6f568-q89r5	1/1	Running	0	12m
oruclosedlooprecovery-568f867b45-b6zld	1/1	Running	0	12m
policymanagementservice-0	1/1	Running	0	12m
rappcatalogueservice-687d69756c-lvwrg	1/1	Running	0	12m

## 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 PMS logs to make sure that the connection between 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 installs the NonRT RIC components. It is simply run like this:

#### Undeploy Nonrtric

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

## Introduction to Helm Chart

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/>