

# Deploy NONRTRIC in Kubernetes

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

- [Demo Video](#)
- [Delivery Status](#)
- [NONRTRIC Archetecture](#)
- [Introduction to Helm Chart](#)
- [Helm Chart in NONRTRIC](#)
- [Packaging & Repo update](#)
- [Deployment](#)
- [Kubernetes Command](#)

## Demo Video

Your browser does not support the HTML5 video element

## Delivery Status



DO NOT EDIT

Only the Integration PTL should edit the delivery status table to track the release validation progress

		OTF	OAM	NONRTRIC	RICP	RICAPP	O-DU	O-CU	Test Result	Notes
Deployment Artifacts	Docker Container									
	Helm Charts									
	Recipe									
E2E Flows	Deployment									
	Undeploy									
	prepare_data.sh									

## NONRTRIC Archetecture

NONRTRIC composes of four major components,

1. Control Panel
2. Policy Management Service
3. A1 Controller
4. NearRT RIC Simulator

Refer [this](#) page for more details

## 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.

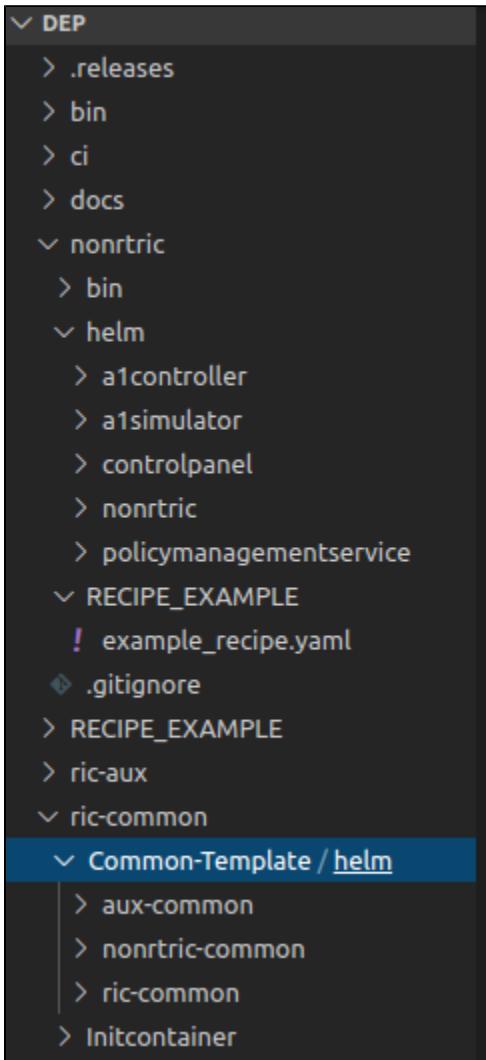
For more information you could refer to below links,

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

# Helm Chart in NONRTRIC

In Release B, the helm chart's are written in [IT/Dep](#) repo.

The following picture shows the hierarchy of the nonrtric within IT/Dep repo.



Each components in nonrtric repo are independent chart with common dependency of nonrtric-common. The nonrtric chart acts a parent chart where other charts like a1controller, a1simulator, controlpanel & policymanagemenservice are added as dependencies in it.

The nonrtric-common contains the common template shared across all the nonrtric componenets. The nonrtric-common chart is added as a dependency into all the components (a1controller, a1simulator, controlpanel, policymanagemenservice & nonrtric)

Each component can be built & packaged independently of others and the required version can be deployed by adding the same into requirements.yaml file in nonrtric

### Requirements.yaml

```
dependencies:
- name: alcontroller
  version: ~1.0.0
  repository: "@local"
- name: alsimulator
  version: ~1.0.1
  repository: "@local"
- name: controlpanel
  version: ~1.0.0
  repository: "@local"
- name: policymanagementservice
  version: ~1.0.0
  repository: "@local"
- name: nonrtric-common
  version: ^1.0.0
  repository: "@local"
```

The values.yaml file contains the default value of the individual components and it can be overridden with the override yaml file. The default values.yaml,

### values.yaml

```
imagePullPolicy: IfNotPresent
image:
  registry: 'nexus3.o-ran-sc.org:10002/o-ran-sc'
  name: nonrtric-al-controller
  tag: 1.7.4
replicaCount: 1
service:
  internalPort: 8181
  externalPort: 8282
```

## Packaging & Repo update

All the logic behind the helm packaging & repo update are kept under deploy-nonrtric.sh file. This script crawls through the nonrtric directory and packages all the dependencies charts & copy it to the helm local repo.

### deploy-nonrtric

```
# Package nonrtric-common and serve it using Helm local repo
HELM_HOME=$(helm home)
COMMON_CHART_VERSION=$(cat $ROOT_DIR/..ric-common/Common-Template/helm/nonrtric-common/Chart.yaml | grep
version | awk '{print $2}')
helm package -d /tmp $ROOT_DIR/..ric-common/Common-Template/helm/nonrtric-common
cp /tmp/nonrtric-common-$COMMON_CHART_VERSION.tgz $HELM_HOME/repository/local/
```

You can use the example recipe or your custom recipe file to invoke this script. Call the below command from /dep/bin directory,

### run deploy command

```
root@vml:~/dep/bin#./deploy-nonrtric -f ../nonrtric/RECIPE_EXAMPLE/example_recipe.yaml
```

The above command packages nonrtric-common, a1controller,a1simulator, controlpanel & policymanagementservice and add them to the local helm repo which will be served later.

The above command also invokes the install script to do the actual deployment of our components which we cover in [Deployment](#) section.

Once you run the deploy command you see the below logs in the console,

#### deploy log

```
Successfully packaged chart and saved it to: /tmp/nonrtric-common-1.0.0.tgz
Packaging NONRTRIC components [controlpanel alcontroller alsimulator policymanagementservice]
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "local" chart repository
...Successfully got an update from the "stable" chart repository
Update Complete. Happy Helming!
Error: Unable to move current charts to tmp dir: rename /root/dep/nonrtric/helm/controlpanel/charts /root/dep/nonrtric/helm/controlpanel/tmpcharts: file exists
Error: found in requirements.yaml, but missing in charts/ directory: nonrtric-common
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "local" chart repository
...Successfully got an update from the "stable" chart repository
Update Complete. Happy Helming!
Saving 1 charts
Downloading nonrtric-common from repo http://127.0.0.1:8879/charts
Deleting outdated charts
Successfully packaged chart and saved it to: /tmp/alcontroller-1.0.0.tgz
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "local" chart repository
...Successfully got an update from the "stable" chart repository
Update Complete. Happy Helming!
Saving 1 charts
Downloading nonrtric-common from repo http://127.0.0.1:8879/charts
Deleting outdated charts
Successfully packaged chart and saved it to: /tmp/alsimulator-1.0.1.tgz
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "local" chart repository
...Successfully got an update from the "stable" chart repository
Update Complete. Happy Helming!
Saving 1 charts
Downloading nonrtric-common from repo http://127.0.0.1:8879/charts
Deleting outdated charts
Successfully packaged chart and saved it to: /tmp/policymanagementservice-1.0.0.tgz
"local" has been removed from your repositories
"local" has been added to your repositories
Finished Packaging NONRTRIC components [controlpanel alcontroller alsimulator policymanagementservice]
```

## Deployment

This is where the actual deployment of the components into kubernetes cluster happens. The install script is responsible for identifying the namespace, helm release & parent chart to start the deployment process. It also creates the config map of the recipe file which is later used during uninstall process. Once all the above are done, it will call the below command to deploy the nonrtric components,

#### install

```
helm install $DIR/../helm/"${PARENT_CHART}" -f ${OVERRIDEYAML} --namespace "${NONRTRIC_NAMESPACE:-nonrtric}" --
name "${RELEASE_PREFIX}"
```

The deploy-nonrtric script is called only the first time and every other redeployment should call install script as long as you don't do any change in helm charts. This saves time and resource as we reuse the existing charts packaged and served by helm repo.

To deploy the nonrtric components in kubernetes you need to call the below command from bin directory within nonrtric,

#### run install command

```
root@vml:~/dep/nonrtric/bin#./install -f ../RECIPE_EXAMPLE/example_recipe.yaml
```

Once the deployment starts you see the below log in console,

## Install log

```
Chart name- nonrtric
namespace/nonrtric created
configmap/nonrtric-recipe created
Deploying NONRTRIC components [controlpanel alcontroller alsimulator policymanagementservice nonrtric]
Updating the Parent Chart [nonrtric]
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "local" chart repository
...Successfully got an update from the "stable" chart repository
Update Complete. Happy Helming!
Saving 5 charts
Downloading alcontroller from repo http://127.0.0.1:8879/charts
Downloading alsimulator from repo http://127.0.0.1:8879/charts
Downloading controlpanel from repo http://127.0.0.1:8879/charts
Downloading policymanagementservice from repo http://127.0.0.1:8879/charts
Downloading nonrtric-common from repo http://127.0.0.1:8879/charts
Deleting outdated charts
NAME: r2-dev-nonrtric
LAST DEPLOYED: Tue Apr 21 10:43:46 2020
NAMESPACE: nonrtric
STATUS: DEPLOYED

RESOURCES:
==> v1/ConfigMap
NAME          DATA   AGE
controlpanel-configmap  1      1s
policymanagementservice-configmap 1      1s

==> v1/Service
NAME           TYPE      CLUSTER-IP     EXTERNAL-IP    PORT(S)        AGE
alcontroller   ClusterIP  10.43.34.119 <none>        8282/TCP      1s
dbhost         ClusterIP  10.43.30.35  <none>        3306/TCP      1s
sdnctldb01    ClusterIP  10.43.89.176 <none>        3306/TCP      1s
al-sim         ClusterIP  None          <none>        8085/TCP      1s
controlpanel   NodePort   10.43.8.241  <none>        8080:30090/TCP 1s
policymanagementservice  NodePort   10.43.79.184 <none>        8081:30092/TCP 1s

==> v1/Deployment
NAME          DESIRED  CURRENT  UP-TO-DATE  AVAILABLE  AGE
alcontroller   1        1        1           0          1s
db             1        1        1           0          1s
controlpanel   1        1        1           0          1s
policymanagementservice  1        1        1           0          1s

==> v1/StatefulSet
NAME          DESIRED  CURRENT  AGE
al-sim-osc   2        1        1s
al-sim-std   2        1        1s

==> v1/Pod(related)
NAME          READY  STATUS        RESTARTS  AGE
alcontroller-5c9f5b586c-wktgg  0/1   ContainerCreating  0          1s
db-549ff9b4d5-9sk15  0/1   ContainerCreating  0          1s
controlpanel-5f4ccc78cb-1c97c  0/1   ContainerCreating  0          1s
policymanagementservice-79d5cdb475-lkhcn  0/1   ContainerCreating  0          1s
al-sim-osc-0  0/1   ContainerCreating  0          1s
al-sim-std-0  0/1   ContainerCreating  0          0s
```

## Kubernetes Command

Once the above process completes without any error, you can check the status of all pod, services, deployments with below commands,

### Get Pods

```
kubectl -n nonrtric get pod
```

NAME	READY	STATUS	RESTARTS	AGE
al-sim-osc-0	1/1	Running	0	150m
al-sim-osc-1	1/1	Running	0	150m
al-sim-std-0	1/1	Running	0	150m
al-sim-std-1	1/1	Running	0	150m
alcontroller-5c9f5b586c-wktgg	1/1	Running	0	150m
controlpanel-5f4ccc78cb-lc97c	1/1	Running	0	150m
db-549ff9b4d5-9sk15	1/1	Running	0	150m
policymanagementservice-79d5cdb475-1khcn	1/1	Running	0	150m

### Get Services

```
kubectl -n nonrtric get services
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
al-sim	ClusterIP	None	<none>	8085/TCP	155m
alcontroller	ClusterIP	10.43.34.119	<none>	8282/TCP	155m
controlpanel	NodePort	10.43.8.241	<none>	8080:30090/TCP	155m
dbhost	ClusterIP	10.43.30.35	<none>	3306/TCP	155m
policymanagementservice	NodePort	10.43.79.184	<none>	8081:30092/TCP	155m
sdnctlDb01	ClusterIP	10.43.89.176	<none>	3306/TCP	155m