



Sterling SkyWave Service Management

Date: 05/31/24

<https://sterling.com/skywave/>

Service Management Overview

The SkyWave Service Management developer extension adds two features to ARC-OTA: Kubernetes Service Orchestration and Service Monitoring.



Software Manifest

Product	Version
Kubernetes	1.26.5-1.1
Containerd	1.6.31-1
Docker CE	26.1.0-1
Calico	3.27.0
MetallB	0.14.3
Helm	3.8.0
Network Operator	23.10.0
GPU Operator	24.3.0 (Driver Version: 535.129.03)
Grafana	10.4.1
Loki	2.9.6
Promtail	2.9.3
Prometheus	2.51.2

Table 1- Service Management v0.4 Software Manifest

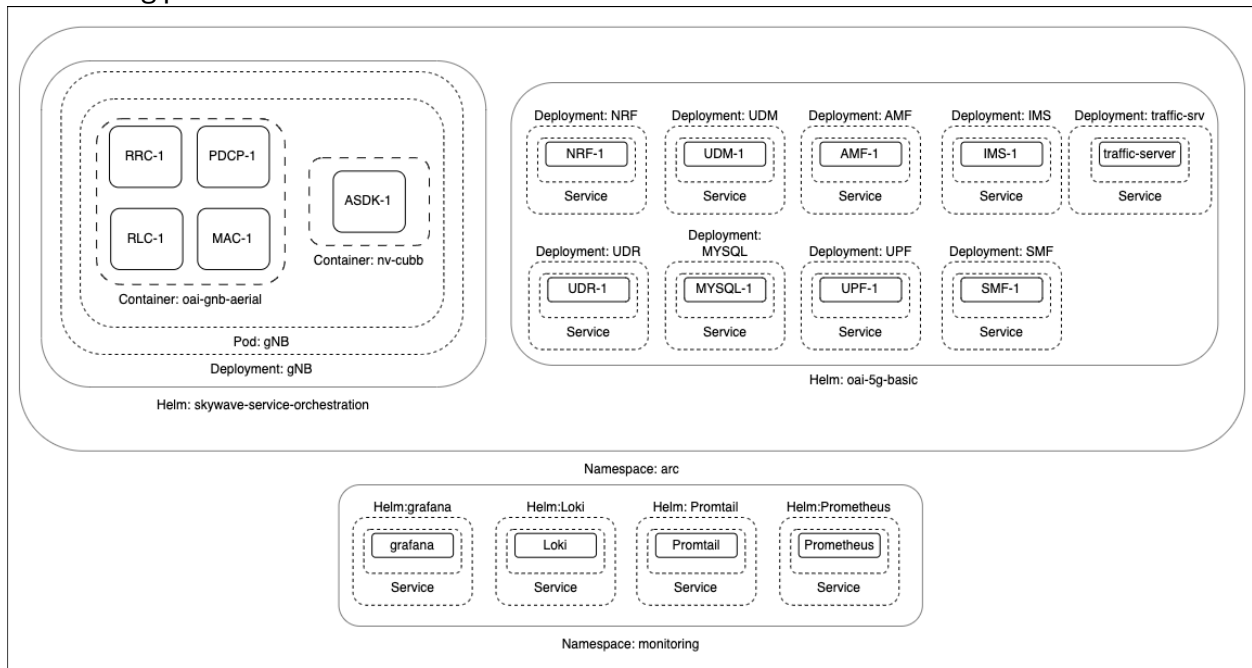
Pod Topologies

Both single-node and multi-node deployment topologies are supported.

Note: In a multi-node topology, each node is installed as a separate Kubernetes single-node cluster.

Single-Node Pod Topology

The single-node topology consists of a single Dell R750 server with the gNB, CN5G and monitoring pods installed.



Dell R750 - (Converged gNB & CN5G)

Figure 1 - Single-Node Pod Topology

Multi-Node Pod Topology

The multi-node topology consists of a dedicated R750 server for the CN5G, Grafana and Loki services and separate R750 servers for each gNB. In a multi-node topology, each gNB has an instance of Prometheus and Promtail which provide data to Grafana and Loki on the CN5G server.

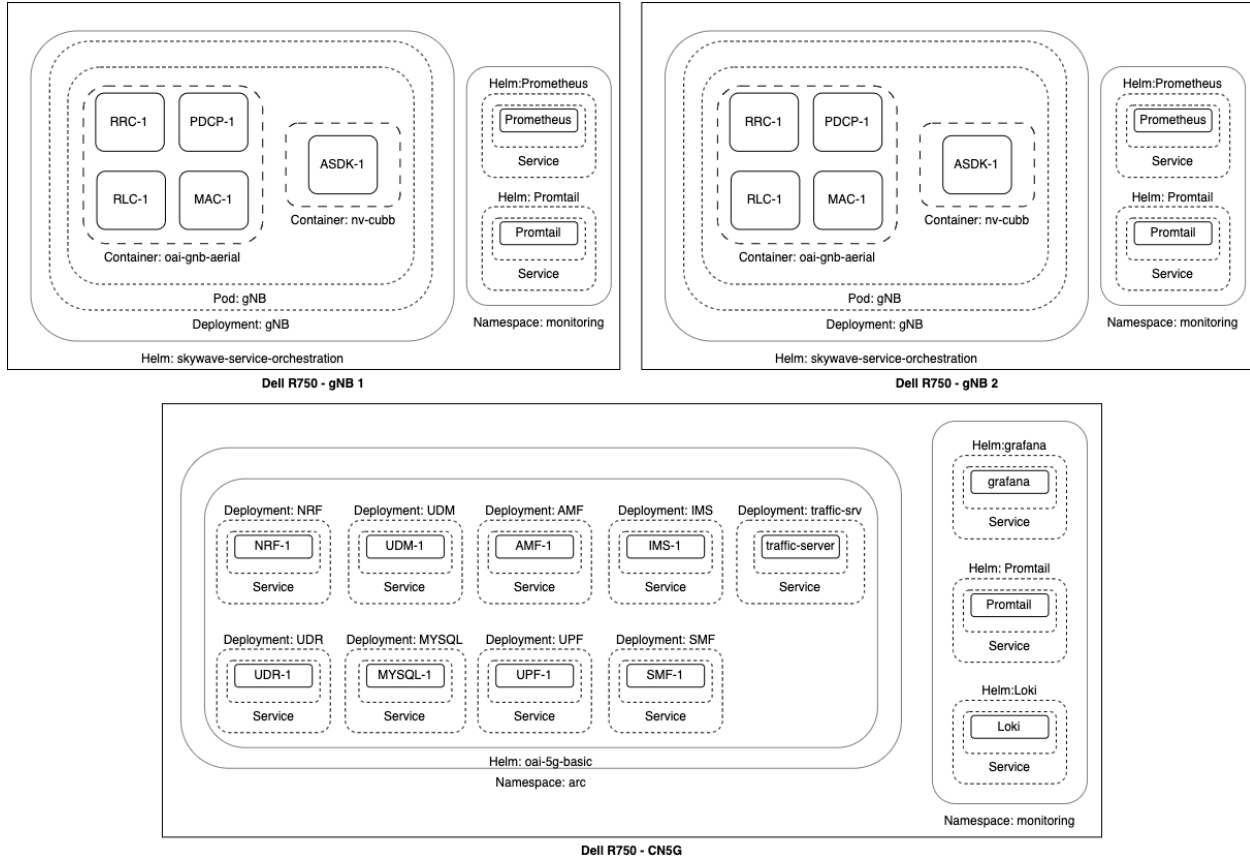


Figure 2 - Multi-Node Pod Topology

Kubernetes Service Orchestration Feature Overview

The developer extension uses Helm to manage applications.

Helm Chart: skywave-service-management

The “skywave-service-management” Helm chart is installed on the gNB server. The Helm chart installs a Kubernetes deployment to create the pod “gnb”. The gnb pod contains the following containers.

- nv-cubb
- oai-gnb-aerial

The nv-cubb container includes the code for Aerial SDK which delivers the Layer 1 capabilities for the gNB. The oai-gnb-aerial container includes code from Open Air Interface (OAI) to deliver Layer 2+ capabilities for the gNB. Both containers are installed in the same pod to provide shared memory between the Layer 1 and Layer 2.

The Helm chart is available for download from NGC. See the Installation Procedure section for download instructions.

Helm Chart: oai-5g-basic

The “oai-5g-basic” Helm chart is installed on the CN5G server in a two-server topology but it can also be installed on the same physical server as the gNB in a single-server topology. This helm chart creates multiple Kubernetes deployments to create the following pods and services.

- asterisk-sip-ims
- oai-5g-basic-mysql
- oai-amf
- oai-ausf
- oai-nrf
- oai-smf
- oai-traffic-server
- oai-udm
- oai-udr
- oai-upf

To expose the oai-amf service to a remote gNB, an additional service is manually created “oai-amf-external”. This service uses the MetalLB load balancer to expose the oai-amf pod using a non-private address.

This Helm chart is available for download from the Eurecom OAI GitLab repository. See the Installation Procedure section for download instructions. The README.md file provides instructions to download and install the Helm chart.

Service Monitoring Feature Overview

The service monitoring feature uses a combination of Grafana, Loki, Promtail and Prometheus. The feature was developed using open-source industry standard tools and it provides a platform that can be extended or modified for each SkyWave end user’s needs.

Included Tools

Grafana

Provides the dashboard user interface which is integrated with the remaining monitoring tools.

Loki

Provides log aggregation and labeling for each log stream.

Promtail

Ships the contents of local filesystem log files to Loki. Promtail provides access to each pod’s log file.

Prometheus

Provides systems monitoring, alerting and a time-series database for historical metrics storage. Prometheus is used to gather the metrics for the GPU and Host dashboards.

Included Dashboards

The service monitoring feature includes three default dashboards.

ARC-OTA Dashboard

- Displays the status of the gNB(s).
- Display the number of UEs per gNB.
- Displays the current throughput of each UE.

GPU Dashboard

- Displays NVIDIA Data Center GPU Manager (DCGM) metrics.
 - GPU Utilization
 - Framebuffer Usage
 - Temperature
 - Clock Speeds
 - Power Usage

Host Dashboards

- Displays an extensive list of CPU, disk, memory and networking metrics.

Installation Procedure

The skywave-service-management Helm package is available for download from NGC.

1. Generate or retrieve your NGC API Key - <https://docs.nvidia.com/ngc/gpu-cloud/ngc-catalog-user-guide/index.html#generating-api-key>.
2. Download, install and configure the NGC CLI - <https://org.ngc.nvidia.com/setup/installers/cli>.
3. *Install Helm*

```
wget https://get.helm.sh/helm-v3.8.0-linux-amd64.tar.gz
tar -zxvf helm-v3.8.0-linux-amd64.tar.gz
sudo mv linux-amd64/helm /usr/local/bin/helm
rm -rf helm-v3.8.0-linux-amd64.tar.gz linux-amd64/
```

4. Download the Helm package. Refer to the ARC-OTA release manifest for the specific version. Use the API key you generated in Step 1.

```
helm fetch https://helm.ngc.nvidia.com/ordaylqpnt3g/charts/skywave-service-management-X.Y.Z.tgz --username='$oauthtoken' --password=YOUR_API_KEY
```

5. Expand the archive.

```
tar -xvzf skywave-service-orchestration-X.Y.Z.tgz  
cd skywave-service-management
```

6. Review the SERVICE_ORCHESTRATION-README.md and SERVICE_MONITORING-README.md files which include instructions on how to deploy the developer extension.