Quick Start Guide

Network and Edge Reference System Architectures - 5G Core UPF

Develop and verify cloud-native services for 5G Core UPF using BMRA on 4th or 5th Gen Intel® Xeon® Scalable processor platform.

Authors

Abhijit Sinha

Introduction

The Reference System Architectures (Reference Systems[1]) are a cloud-native, forward-looking Kubernetes*-cluster template solution for network implementations. They provide Ansible* playbooks that define configuration profiles for fast, automatic deployment of needed cluster services and capabilities.

This document is a quick start guide to configure the **Container Bare Metal Reference System Architecture (BMRA)** on **4th or 5th Gen Intel® Xeon® Scalable processor**-based platform for **5G Core User Plane Function (UPF).**

The Reference System has a variety of configuration profile settings for different network traffic workloads. **This quick start guide enables 5G UPF use case using the Remote Central Office-Forwarding Configuration Profile**. For details on this and other Configuration Profiles and hardware options, refer to the User Guides listed in the Reference Documentation section.

Remote Central Office-Forwarding Configuration Profile Architecture

Figure 1 shows the architecture of the Remote Central Office-Forwarding (remote_fp) profile, which deploys the infrastructure for 5G Core.

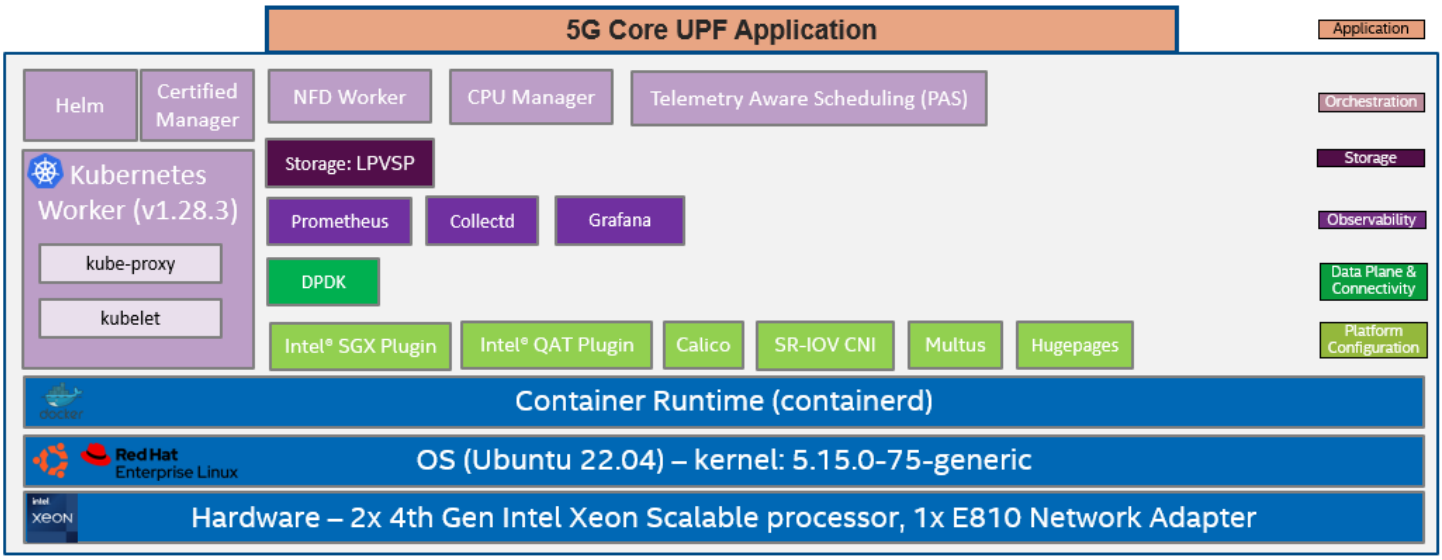

Figure 1: Remote Central Office-Forwarding Configuration Profile Architecture

[1] In this document, "Reference System" refers to the Network and Edge Reference System Architecture.

Hardware BOM

Following is the list of the hardware components that are required for setting up reference systems:

Ansible host	Laptop or server running a UNIX base distribution
Controller Node(s)	Any 4th or 5th Gen Intel® Xeon® Scalable processor-based server
Worker Node(s)	1x 4th or 5th Gen Intel® Xeon® Scalable processors on Intel® SDP S2EG4SEQ5Q
Ethernet Adapter	Intel® Ethernet Network Adapter E810-CQDA2 or Intel® Ethernet Controller XXV/XL710
Recommended BIOS	"Max Performance Turbo" BIOS configuration (refer to Chapter 3.8 of the BMRA User Guide) - Enable SGX in the BIOS

Software BOM

Following is the list of the software components that are required for setting up reference systems:

Security	OpenSSL, Intel® SGX, Intel® QAT
Storage	LPVSP
Observability	Prometheus, Collectd, Grafana
Acceleration/ Data Plane	DPDK
Operators & Device plugins	Intel® QAT and Intel® SGX plugins, Multus, SRIOV network operator, Intel® Ethernet Operator
Container Runtime	containerd
Orchestration	Kubernetes v1.28.3, Assisted Scheduling: Telemetry Aware Scheduling (TAS)
OS	Ubuntu 22.04.2 LTS (kernel 5.15.0-72 generic) and RHEL 9.2

For more details of software versions for the **Remote Central Office-Forwarding Configuration Profile**, refer to Chapter 4 of the BMRA User Guide listed in the Reference Documentation section.

Intel® DLB, Intel® DSA, and Intel® Scalable IOV SW are set as optional in the **Remote Central Office-Forwarding Configuration Profile**.

Getting Started

Pre-Requisites

Before starting the deployment, perform the following steps:

- A fresh OS installation is expected on the controller and target nodes to avoid a conflict between the RA deployment process with the existing software packages. To deploy RA on the existing OS, ensure that there is no prior Docker or Kubernetes* (K8s) installations on the server(s).
- The controller and target server hostname(s) must be in lowercase, numerals, and hyphen '-'.
    - For example: wrk-8 is acceptable, wrk_8, WRK8, Wrk^8 are not accepted as hostnames.
- The servers in the cluster are Network Time Protocol (NTP) synced, i.e., they must have the same date and time.
- The BIOS on the target server is set as per the recommended settings.

Deployment Setup

Figure 2 shows the deployment model for Remote Central Office-Forwarding Configuration using BMRA. The Ansible host is used for configuring and deploying BMRA on a set of target servers.

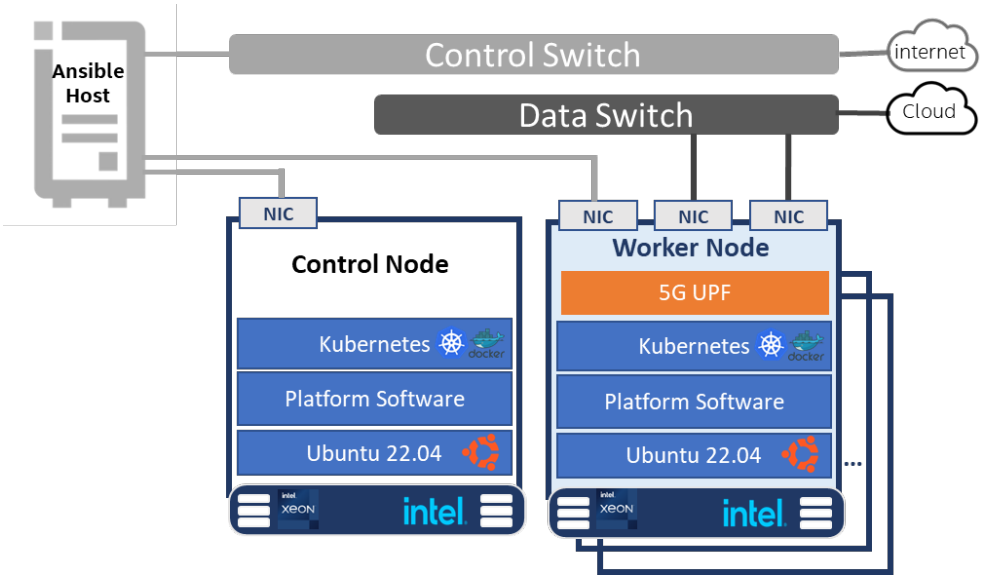

Figure 2: BMRA deployment set-up for 5G Core UPF

Note: The K8s cluster deployed using BMRA is scalable to multiple controller and worker nodes.

Installation Flow for RA Deployment

Ansible playbooks are used to install the Bare Metal Reference Systems Architecture (BMRA). Before the playbooks can be run, there are a few steps to prepare the environment and change relevant configuration options.

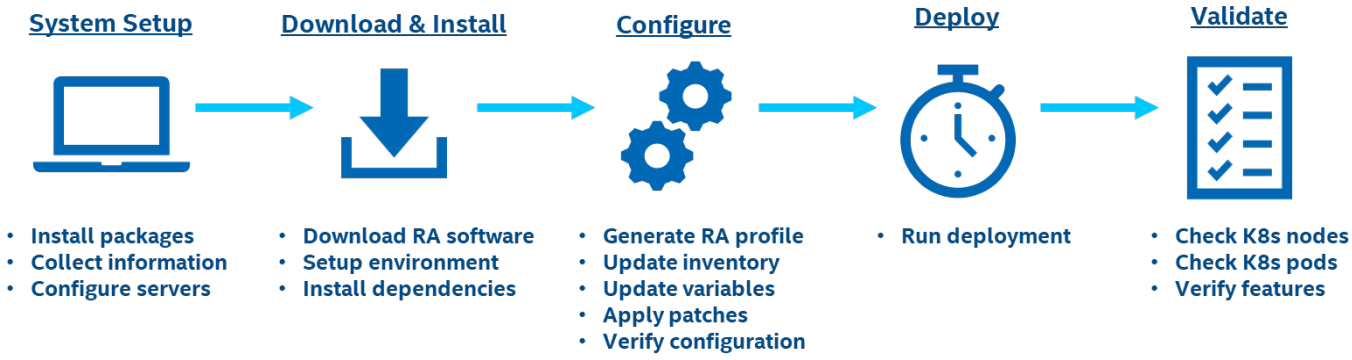

Figure 3: RA Deployment flow

Step 1 - Set Up the System

The below mentioned steps assume that both the Ansible host and target servers are running Ubuntu as the operating system and the user has root or sudo access. For RHEL, use 'yum' or 'dnf' as the package manager instead of 'apt'.

System Setup

Ansible Host

1. Install necessary packages (some might already be installed):

```
# sudo apt update
# sudo apt install -y python3 python3-pip openssh-client git build-essential
# pip3 install --upgrade pip
```

2. Generate a SSH keypair if needed (check /root/.ssh/):

```
# ssh-keygen -t rsa -b 4096 -N "" -f ~/.ssh/id_rsa
```

3. Copy the public key to the target servers - controller and worker nodes:
```
# ssh-copy-id root@<target IP>
```
4. Verify passwordless connectivity to the target servers:
```
# ssh root@<target IP>
```

Target Server

1. Install necessary packages (some might already be installed):
```
# sudo apt install -y python3 openssh-server lshw
```
2. As part of the configuration in Step 3, information about PCI devices for SR-IOV must be specified. Find the relevant PCI IDs (bus:device.function) using 'lspci', and note down the IDs for later when configuring host_vars on the Ansible host:
```
# lspci | grep Eth
18:00.0 Ethernet controller: Intel Corporation Ethernet Controller E810-C for QSFP (rev 01)
18:00.1 Ethernet controller: Intel Corporation Ethernet Controller E810-C for QSFP (rev 01)
```
3. Verify if the target server's CPU has an on-chip QAT device.
```
# lspci -nnD | grep 494*
0000:76:00.0 Co-processor [0b40]: Intel Corporation Device [8086:4942] (rev 40)
0000:7a:00.0 Co-processor [0b40]: Intel Corporation Device [8086:4942] (rev 40)
0000:f3:00.0 Co-processor [0b40]: Intel Corporation Device [8086:4942] (rev 40)
0000:f7:00.0 Co-processor [0b40]: Intel Corporation Device [8086:4942] (rev 40)
```

Step 2 - Download and Install

Ansible Host

1. Download the source code from GitHub repository for the Reference System server:
```
# git clone https://github.com/intel/container-experience-kits/
# cd container-experience-kits
# git checkout v24.01
```
2. Set up Python* virtual environment and install dependencies:
```
# python3 -m venv venv
# source venv/bin/activate
# pip3 install -r requirements.txt
```
3. Install Ansible dependencies for the Reference System:
```
ansible-galaxy install -r collections/requirements.yml
```

Download & Install

Step 3 - Configure BMRA

Configure

The **Remote Central Office-Forwarding** configuration profile (remote_fp) is used for 5G UPF deployment.

Ansible Host

1. Generate the configuration files:
```
# make k8s-profile PROFILE=remote_fp ARCH=spr
```
2. Update the **inventory.ini** file to match the deployment set-up. The values for *<target hostname>* and *<target IP>* must be updated to match the target systems in the BMRA cluster as shown in Figure 2 .
```
# vim inventory.ini
[all]
<controller hostname> ansible_host=<controller IP> ip=<controller IP> ansible_user=root
<worker hostname>     ansible_host=<worker IP> ip=<worker IP> ansible_user=root
localhost             ansible_connection=local ansible_python_interpreter=/usr/bin/python3

[vm_host]

[kube_control_plane]
<controller hostname>

[etcd]
<controller hostname>

[kube_node]
```

```
<worker hostname>

[k8s_cluster:children]
kube_control_plane
kube_node

[all:vars]
ansible_python_interpreter=/usr/bin/python3
```

3. Update the host_vars filename with the target machine's hostname:

```
# mv host_vars/node1.yml host_vars/<worker hostname>.yml
```

To utilize features depending on SR-IOV and Intel® QAT, host_vars must be updated with information about the PCI devices on the worker node server. The example below can be used as a reference for the configuration but should be updated to match the correct PCI IDs of the worker node server.

4. Update *host_vars/<worker hostname>.yml* with PCI device information specific to the worker node server:

```
# vim host_vars/<worker hostname>.yml
dataplane_interfaces:
  - bus_info: "18:00.0"                          # Use the SR-IOV PCI ID here
    pf_driver: ice
    ddp_profile: "ice_comms-1.3.40.5.pkg"
    default_vf_driver: "iavf"
    sriov_numvfs: 8
    sriov_vfs:                                   # This is optional but can be used to change
the driver of specific VFs. Any VF not specified here will use the "default_vf_driver"
specified above.
      vf_00: "vfio-pci"
      vf_05: "vfio-pci"
```

Note: Additional details about the configuration options and values can be found as comments in the file.

Note: Be sure to remove the square brackets [ ] that follows the 'dataplane_interfaces' configuration option by default.

5. If the server is behind a proxy, update *group_vars/all.yml* by updating and uncommenting the lines for http_proxy, https_proxy, and additional_no_proxy.

```
# vim group_vars/all.yml
## Proxy configuration ##
http_proxy: "http://proxy.example.com:port"
https_proxy: "http://proxy.example.com:port"
additional_no_proxy: ".example.com,mirror_ip"
```

6. (Required) Apply required patches for Kubespray:

```
# ansible-playbook -i inventory.ini playbooks/k8s/patch_kubespray.yml
```

7. (Optional, recommended) Verify that Ansible can connect to the target servers, by running the below command and checking the output generated in the **all_system_facts.txt** file:

```
# ansible -i inventory.ini -m setup all > all_system_facts.txt
```

8. (Optional, recommended) Check dependencies of components enabled in group_vars and host_vars with the packaged dependency checker. This step is also run by default as part of the main playbook:

```
# ansible-playbook -i inventory.ini playbooks/preflight.yml
```

Step 4 - Deploy

Ansible Host

Now the Reference System cluster can be deployed using the below command:

Deploy

```
# ansible-playbook -i inventory.ini playbooks/remote_fp.yml --flush-cache
```

Note: If the playbook fails or if you want to clean up the environment to run a new deployment, you can optionally use the provided Cluster Removal Playbook to remove any previously installed Kubernetes and related plugins.

```
# ansible-playbook -i inventory.ini playbooks/redeploy_cleanup.yml
```

Step 5 - Validate

Ansible Host

Validate

1. To interact with the Kubernetes CLI (kubectl), start by connecting to a controller node in the cluster:

```
# ssh root@<controller-node ip>
```

2. Once connected, the status of the Kubernetes cluster can be checked:

```
# kubectl get nodes -o wide
# kubectl get pods --all-namespaces
```

Additional feature verification tests can be found here:
https://github.com/intel/container-experience-kits/tree/master/validation/verification-manual

Reference Documentation

The *Network and Edge Bare Metal Reference System Architecture User Guide* provides information and full set of installation instructions for a BMRA.

The *Network and Edge Virtual Machine Reference System Architecture User Guide* provides information and full set of installation instructions for a VMRA.

The *Network and Edge Cloud Reference System Architecture User Guide* provides the means to develop and deploy cloud-native applications in a CSP environment and still experience Intel® technology benefits.

The *Network and Edge Reference System Architectures Portfolio User Manual* provides additional information for the Reference System including a complete list of reference documents.

Other collaterals, including technical guides and solution briefs that explain in detail the technologies enabled in the Reference Systems are available in the following location: Network & Edge Platform Experience Kits.

Document Revision History

REVISION	DATE	DESCRIPTION
001	July 2023	Initial release.
002	October 2023	Updated BMRA version to 23.10, added profile architecture and Ansible flow for deployment.
003	January 2024	Updated BMRA version to 24.01.

0124/DN/WIT/PDF 784313-003US