User Guide

Network and Edge Container Bare Metal Reference System Architecture Release v24.01

Authors

Aparna Balachandran

Octavia Carotti

Alek Du

Veronika Karpenko

Zhifang Long

Dana Nehama

Abhijit Sinha

Mathieu Sobrero

Daniel Ugarte

1 Introduction

1.1 User Guide Information

The Container Bare Metal Reference Architecture (BMRA) is part of the Network and Edge Reference System Architectures (Reference System[1]) Portfolio. The BMRA is a cloud-native, forward-looking, common-template platform for network implementations. It addresses the need to deploy bare metal systems and cloud-native Kubernetes* clusters, optimized with Intel® hardware and software innovation for diverse workloads across network locations.

This user guide provides a comprehensive description of the BMRA Release v24.01 deployment and verification processes. By following this document, it is possible to set up automatically, using Ansible* playbooks, a complete containers bare metal system. The document contains installation and configuration instructions, including the use of BIOS options, multiple operating systems, Kubernetes, platform software features, and device plug-ins. The document also goes into detail about the open-source Ansible playbooks that automatically provision the BMRA. Use case and single server step-by-step instructions are provided in quick start guides.

1.2 Purpose and Scope

Services delivered across the network require the deployment of different hardware, software, and configuration specifications due to varying cost, density, and performance requirements. The BMRA common platform allows support for these diverse deployment needs using Network Location Configuration Profiles. In addition, a generic, “a-la-cart” Configuration Profile is available for engineers who need the flexibility to build, modify, and evaluate diverse BMRA software options. Ansible playbooks automatically implement the Configuration Profiles for fast and predicted deployment. The result is an installed, optimized Reference System Architecture Flavor as specified by the Configuration Profile.

1.3 Configuration Profiles and Quick Start Guides

Network Location Configuration Profiles covered in this document include:

- **On-Premises Edge Configuration Profile** – Typical customer premises deployment.
- **On-Premises VSS Configuration Profile** – Customer premises deployment supporting Video Structuring Server (VSS).
- **On-Premises AI Box Configuration Profile** – Customer premises deployment supporting Intel® Edge AI Box.
- **On-Premises SW Defined Factory Configuration Profile** – Industrial deployment.
- **Access Edge Configuration Profile** – Far edge wireless-access network deployments, tuned to support virtual radio access network (vRAN) and FlexRAN™ solution deployments, which require high throughput, low latency, security, and power management control.
- **Remote Central Office-Forwarding Configuration Profile** – Near edge deployments supporting fast packet-forwarding workloads such as cable modem termination system (CMTS), user plane function (UPF), and application gateway function (AGF).
- **Regional Data Center Configuration Profile** – Central-office location typical Configuration Profile, tailored for video production and visual processing workloads such as CDN transcoding.

Generic Configuration Profiles enable flexible deployments and include the following:

- **Basic Configuration Profile** – A generic minimum BMRA Kubernetes cluster setup.
- **Build-Your-Own Configuration Profile** – A BMRA Kubernetes cluster setup allowing you to select your preferred options.

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

Quick Start Guide	Configuration Profile	Hardware
Network and Edge Reference System Architectures - Single Server Quick Start Guide	▪ Basic ▪ Build-Your-Own	▪ 4th and 5th Gen Intel® Xeon® Scalable processors and Intel® Xeon® D processors ▪ Intel® Core™ processors ▪ Intel Atom® processors
Network and Edge Reference System Architectures - CDN Quick Start Guide	▪ On-Premises Edge	▪ 4th and 5th Gen Intel® Xeon® Scalable processors and Intel® Xeon® D processors
Network and Edge Reference System Architectures - Edge Analytics Video Structuring Server (VSS) Quick Start Guide	▪ On-Premises VSS	▪ 4th and 5th Gen Intel® Xeon® Scalable processors
Network and Edge Reference System Architectures - On Premises Edge AI Box Quick Start Guide	▪ On-Premises AI Box	▪ 12th Gen Intel® Core™ processor ▪ 13th Gen Intel® Core™ processors ▪ Intel® Core™ Ultra processors
Network and Edge Reference System Architectures - Industrial Controller Quick Start Guide	▪ On-Premises SW Defined Factory	▪ 12th Gen Intel® Core™ processors ▪ Intel Atom® processor x6000 series
Network and Edge Reference System Architectures - vRAN Setup with FlexRAN™ Software Quick Start Guide	▪ Access Edge	▪ 4th Gen Intel® Xeon® Scalable processors ▪ 4th Gen Intel® Xeon® Scalable processors with Intel® vRAN Boost
Network and Edge Reference System Architectures - 5G vRAN Security Quick Start Guide	▪ Access Edge	▪ 4th Gen Intel® Xeon® Scalable processors ▪ 4th Gen Intel® Xeon® Scalable processors with Intel® vRAN Boost
Network and Edge Reference System Architectures - 5G Core UPF Quick Start Guide	▪ Remote Central Office-Forwarding	▪ 4th and 5th Gen Intel® Xeon® Scalable processors and Intel® Xeon® D processors
Network and Edge Reference System Architectures - Video Production Quick Start Guide	▪ Regional Data Center	▪ 4th and 5th Gen Intel® Xeon® Scalable processors and Intel® Xeon® D processors

More information on Configuration Profiles is provided later in this document.

1.4 Version 24.01 Release Information

BMRA v24.01 common platform supports 4th and 5th Gen Intel® Xeon® Scalable processors, Intel® Xeon® D processors, Intel® Core™ processors, and Intel Atom® processors. Other advanced Intel® hardware technologies supported include the Intel® Ethernet Controller, Intel® QuickAssist Technology (Intel® QAT), Intel® Server GPU, Intel® Data Center GPU Flex Series, Intel® FPGA SmartNIC WSN6050 Platform, and Intel® Infrastructure Processing Unit (Intel® IPU) ASIC E2000 card.

The supported software components comprise open-source cloud-native software delivered by Intel, partners, and open-source communities (for example, Kubernetes, Telegraf, Istio, FD.io).

Release v24.01 builds upon prior release. The following are the key updates. For details, see Reference System Release Notes.

Use Case Updates:

- Support for Intel® Edge AI Box 4.0 on 12th and 13th Gen Intel® Core™ and Intel® Core™ Ultra processors
- Support for Intel® In-Band Manageability framework software to enable an administrator to perform critical device management operations for Intel® Edge AI box.
- Support for Intel ® Workload Services Framework version of the Istio* Envoy* workload
- Support VMs provisioned via KubeVirt* (v1.1.0)
- Support external access to services via Ingress controller for Kubernetes using NGINX* as a reverse proxy and load balancer

Hardware Updates:

- Intel® Core™ Ultra Processor for Edge deployments
- Support for 5th Gen Intel® Xeon® Scalable processor for the Edge/IoT

Software Updates: Software versions upgraded for the majority of Reference System components (See Section 4.1)

Under NDA:

Select capabilities available under NDA are integrated and validated with the BMRA. Contact your Intel representative for access to the following NDA material:

- FlexRAN™ software v23.11
- Intel® QuickAssist Technology (Intel® QAT) drivers for 5th Gen Intel® Xeon® Scalable processors
- Intel® AI Boost driver for Intel® Core™ Ultra processor
- Intel® Core™ Ultra processor audio firmware and topology

Experience Kits, the collaterals that explain in detail the technologies enabled in BMRA Release 24.01, including benchmark information, are available on Intel Network Builder at Network & Edge Platform Experience Kits.

Table of Contents

Figures

Tables

Document Revision History

Three previous editions of the BMRA document were released, starting April 2019.

- Covered 2nd Gen Intel® Xeon® Scalable processors
- Covered 2nd and 3rd Gen Intel® Xeon® Scalable processors
- Covered 2nd and 3rd Gen Intel® Xeon® Scalable processors and Intel® Xeon® D processor

Revision	Date	Description
001	February 2022	Initial release.
002	March 2022	Updated a few URLs.
003	June 2022	Covers the 4th Gen Intel® Xeon® Scalable processor (formerly code named Sapphire Rapids). See "Version 22.05 Release Information" for details.
004	June 2022	Changes include updates to the discussion of the BMRA for Storage Deployment Model.
005	July 2022	Added NDA support for FlexRAN™ software, updated Istio and service mesh features.

Revision	Date	Description
006	August 2022	The changes include updates to the discussions of the Access Edge Configuration Profile and Intel® Ethernet Operator.
007	October 2022	Updated for BMRA Release 22.08; added information about the new Cloud Reference System Architecture (Cloud RA) deployment model.
008	December 2022	Updated for BMRA Release 22.11; includes improvements and updates on Reference System in alignment with the launch of the 4th Gen Intel® Xeon® Scalable processor.
009	March 2023	Updated for BMRA Release 23.02; includes improvements to run FlexRAN™ software in a container and addition of Media Analytics Libraries.
010	July 2023	Updated for BMRA Release 23.07; includes support for 5th Gen Intel® Xeon® Scalable processors, Intel® Core™ processors, and Intel Atom® processors and other hardware/software updates.
011	October 2023	Updated for BMRA Release 23.10; includes improvements and updates for use cases as well as alignment with the launch of the 5th Gen Intel® Xeon® Scalable processor.
012	January 2024	Updated for BMRA Release 24.01; includes improvements and updates and support for Intel® Core™ Ultra processors, Intel® AI Box 4.0, Intel® In-Band Manageability framework, and VM provisioning via KubeVirt*.

1.5 Terminology

Table 1 lists the key terms used throughout the portfolio. These terms are specific to Network and Edge Reference System Architectures Portfolio deployments.

Table 1. Terminology

Abbreviation	Description
Experience Kits	Guidelines delivered in the form of—manuals, user guides, application notes, solution briefs, training videos—for best-practice implementation of cloud native and Kubernetes technologies to ease developments and deployments.
Network and Edge Reference System Architectures Portfolio	A templated system-level blueprint for a range of locations in enterprise and cloud infrastructure with automated deployment tools. The portfolio integrates the latest Intel platforms and cloud-native technologies for multiple deployment models to simplify and accelerate deployments of key workloads across a service infrastructure.
Deployment Model	Provides flexibility to deploy solutions according to business and IT needs. The portfolio offers three deployment models: ▪ **Container Bare Metal Reference System Architecture (BMRA)** – A deployment model of a Kubernetes cluster with containers on a bare metal platform. ▪ **Virtual Machine Reference System Architecture (VMRA)** – A deployment model of a virtual cluster on a physical node. The virtual cluster can be a Kubernetes containers-based cluster. ▪ **Cloud Reference System Architecture (Cloud RA)** – A deployment model that uses CSP's Intel-based instances for running cloud-native applications in the Cloud. The worker instances are provided based on the Configuration Profile that workload demands.
Configuration Profiles	A prescribed set of components—hardware, software modules, hardware/software configuration specifications—designed for a deployment for specific workloads at a network location (such as Access Edge). Configuration Profiles define the components for optimized performance, usability, and cost per network location and workload needs[2] In addition, generic Configuration Profiles are available for developers' flexible deployments.
Reference System Architecture Flavor	An instance of reference architecture generated by implementing a Configuration Profile specification.
Ansible Playbook	A set of validated scripts that prepare, configure, and deploy a Reference System Architecture Flavor per Configuration Profile specification.
Configuration Profile Ansible Scripts	Automates quick, repeatable, and predictive deployments using Ansible playbooks. Various Configuration Profiles and Ansible scripts allow automated installations that are application-ready, depending on the workload and network location.
Kubernetes Cluster	A deployment that installs at least one worker node running containerized applications. Pods are the components of the application workload that are hosted on worker nodes. Control nodes manage the pods and worker nodes.
Intel Platforms	Prescribes Intel platforms for optimized operations. The platforms are based on 4th and 5th Gen Intel® Xeon® Scalable processors plus the Intel® Xeon® D processor. These platforms include the Taylors Falls Reference Design. The platforms integrate Intel® Ethernet Controller 700 Series and 800 Series, Intel® QuickAssist Technology (Intel® QAT), Intel® Server GPU (graphics processing unit), Intel® Optane™ technology, and more.

In addition to key terms, portfolio deployment procedures follow a hardware and software configuration taxonomy. Table 2 describes the taxonomy used throughout this document.

Table 2. Hardware and Software Configuration Taxonomy

Term	Description
Hardware Taxonomy	
ENABLED	Setting must be enabled in the BIOS (configured as Enabled, Yes, True, or similar value)
DISABLED	Setting must be disabled in the BIOS (configured as Disabled, No, False, or any other value with this meaning.)
OPTIONAL	Setting can be either disabled or enabled, depending on workload. Setting does not affect the Configuration Profile or platform deployment
Software Taxonomy	
TRUE	Feature is included and enabled by default
FALSE	Feature is included but disabled by default - can be enabled and configured by user
N/A	Feature is not included and cannot be enabled or configured

[2] Workloads and configurations. Results may vary.

1.6 Intel Investments of Capabilities

Intel investments in networking solutions are designed to help IT centers accelerate deployments, improve operational efficiencies, and lower costs. Table 3 highlights Intel investments in the portfolio and their benefits.

Table 3. Intel Capabilities Investments and Benefits

Capability	Benefit
Performance	Intel® platform innovation and accelerators, combined with packet processing innovation for cloud-native environments, deliver superior and predictive application and network performance.
Orchestration and Automation	Implementing Kubernetes containers orchestration, including Kubernetes Operators, simplifies and manages deployments and removes barriers in Kubernetes to support networking functionality.
Observability	Collecting platform metrics by using, as an example, the collectd daemon and Telegraf server agent, publishing the data, and generating reports, enables high visibility of platform status and health.
Power Management	Leveraging Intel platform innovation, such as Intel® Speed Select Technology (Intel® SST), supports optimized platform power utilization.
Security	Intel security technologies help ensure platform and transport security. These technologies include the following: ▪ Intel® Security Libraries for Data Center (Intel® SecL - DC) ▪ Intel® QuickAssist Technology Engine for OpenSSL (Intel® QAT Engine for OpenSSL) ▪ Intel® Software Guard Extensions (Intel® SGX) ▪ Intel® Trust Domain Extensions (Intel® TDX) on 5th Gen Intel® Xeon® Scalable processors only ▪ Key Management Reference Application (KMRA) implementation
Storage	Creating a high-performance, scalable local-storage or remote-storage platform using diverse storage technologies (Object Storage; File/Block storage) and implementations. For example, MinIO implementation for remote Object Storage supports data-intensive applications, such as media streaming, big data analytics, AI, and machine learning.
Service Mesh	Implementing a service mesh architecture using Istio allows application services that can be added, connected, monitored, more secure, and load-balanced with few or no code changes. Service mesh is integrated with the Trusted Certificate Service for Kubernetes* platform, providing more secure key management.

1.7 Reference Documentation

The Network and Edge Reference System Architectures Portfolio User Manual contains a complete list of reference documents. A virtual machine-based reference architecture (VMRA) deployment allows creation of a Kubernetes cluster for a Configuration Profile on a virtualized infrastructure. The Network and Edge Virtual Machine Reference System Architecture User Guide provides information and installation instructions for a VMRA. The Cloud Reference System Architecture (Cloud RA) provides the means to develop and deploy cloud-native applications in a CSP environment and still experience Intel® technology benefits. Find more details in the Network and Edge Cloud Reference System Architecture User Guide.

Access quick start guides for step-by-step instructions to start building the BMRA directly.

- Network and Edge Reference System Architectures - Single Server Quick Start Guide
- Network and Edge Reference System Architectures - Edge Analytics Video Structuring Server (VSS) Quick Start Guide
- Network and Edge Reference System Architectures - Industrial Controller Quick Start Guide
- Network and Edge Reference System Architectures - vRAN Setup with FlexRAN™ Software Quick Start Guide
- Network and Edge Reference System Architectures - 5G vRAN Security Quick Start Guide
- Network and Edge Reference System Architectures - 5G Core UPF Quick Start Guide
- Network and Edge Reference System Architectures - CDN Quick Start Guide
- Network and Edge Reference System Architectures - Video Production Quick Start Guide
- Network and Edge Reference System Architectures - On Premises Edge AI Box Quick Start Guide

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

2 Reference Architecture Deployment

This chapter explains how a Reference System Architecture Flavor is generated and deployed. The process includes installation of the hardware setup followed by system provisioning.

2.1 BMRA Architecture

The BMRA is a Kubernetes cluster that can be configured to support a flexible number of Kubernetes control nodes and worker nodes (see Figure 1). To deploy the BMRA, you deploy and configure the following elements:

- **Hardware Components**: Multiple platform hardware options are available, including a variety of 5th and 4th Gen Intel® Xeon® Scalable processor SKUs, Intel® Xeon® D processor SKUs, Intel® Ethernet Network Adapters, Intel® QAT, and Intel® Server GPUs. BIOS options are listed elsewhere in this guide. Deployment engineers should refer to Section 3.7 during deployment to select and configure optimal BIOS values before cluster provisioning.
- **Software Capabilities**: The software capabilities are based on open-source software delivered by cloud-native and CNCF communities driving Kubernetes, Istio, observability, DPDK, FD.io. OVS, OVS-DPDK, and through Intel GitHub. Options for RHEL and Ubuntu* Linux* operating systems are available. The container environment is based on Docker, containerd, or CRI-O container runtimes.
- **Configuration Profiles**: Specific hardware and software configurations are provided in the Configuration Profiles based on Intel assessment and verification.
- **Installation Playbooks**: Ansible playbooks implement the Configuration Profiles for best-practice, reliable, and accelerated Reference System Architecture Flavor deployment.

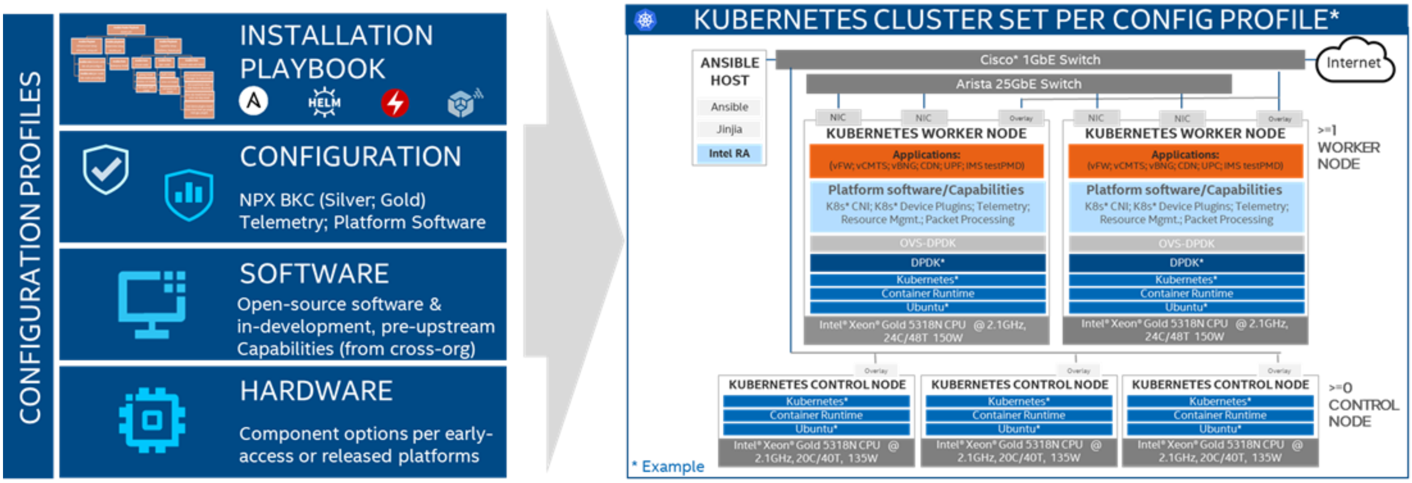
CONFIGURATION PROFILES
INSTALLATION PLAYBOOK
HELM
CONFIGURATION
NPX BKC (Silver; Gold)
Telemetry; Platform Software
SOFTWARE
Open-source software &
in-development, pre-upstream
Capabilities (from cross-org)
HARDWARE
Component options per early-access or released platforms
KUBERNETES CLUSTER SET PER CONFIG PROFILE*
ANSIBLE HOST
Ansible
Jinjia
Intel RA
Cisco* 1GbE Switch
Internet
Arista 25GbE Switch
NIC
NIC
Overlay
KUBERNETES WORKER NODE
Applications:
(vFW; vCMTS; vBNG; CDN; UPF; IMS testPMD)
Platform software/Capabilities
K8s* CNI; K8s* Device Plugins; Telemetry; Resource Mgmt.; Packet Processing
OVS-DPDK
DPDK*
Kubernetes*
Container Runtime
Ubuntu*
Intel® Xeon® Gold 5318N CPU @ 2.1GHz, 24C/48T 150W
NIC
NIC
Overlay
KUBERNETES WORKER NODE
Applications:
(vFW; vCMTS; vBNG; CDN; UPC; IMS testPMD)
Platform software/Capabilities
K8s* CNI; K8s* Device Plugins; Telemetry; Resource Mgmt.; Packet Processing
OVS-DPDK
DPDK*
Kubernetes*
Container Runtime
Ubuntu*
Intel® Xeon® Gold 5318N CPU @ 2.1GHz, 24C/48T 150W
>=1 WORKER NODE
Overlay
KUBERNETES CONTROL NODE
Kubernetes*
Container Runtime
Ubuntu*
Intel® Xeon® Gold 5318N CPU @ 2.1GHz, 20C/40T, 135W
Overlay
KUBERNETES CONTROL NODE
Kubernetes*
Container Runtime
Ubuntu*
Intel® Xeon® Gold 5318N CPU @ 2.1GHz, 20C/40T, 135W
Overlay
KUBERNETES CONTROL NODE
Kubernetes*
Container Runtime
Ubuntu*
Intel® Xeon® Gold 5318N CPU @ 2.1GHz, 20C/40T, 135W
>=0 CONTROL NODE
* Example

Figure 1. BMRA Illustration and Applicable Elements

2.2 Configuration Profiles

A Configuration Profile describes specific hardware and software bills of material (BOMs) and configurations, applicable for a specific deployment. Configuration Profiles consider the best-known configuration (BKC) validated by Intel for optimized performance.[3]

Installation scripts are available to deploy the required components for a Reference System Architecture Flavor. Each BMRA is built on the following:

- **Intel Platform foundation** with Intel processors and technologies.
- **Hardware BOM** optimized for delivering an application at a specific location using a deployment model. For example, to support a UPF workload at the Remote CO, the BMRA deployment is populated with the maximum available Ethernet adapters or network interface cards (NICs).
- **Software BOM** leverages the Intel platform and enables cloud-native adoption.
- **Installation (Ansible) Playbook** automates the installation of a Reference System Architecture Flavor per a Configuration Profile specification.

The following Reference Architecture Configuration Profiles are network location-specific:

- **Remote Central Office-Forwarding Configuration Profile** – Clusters ranging from a half rack to a few racks of servers, typically in a pre-existing, repurposed, unmanned structure. The usage scenarios include running latency-sensitive

[3] Workloads and configurations. Results may vary.

applications near the user (for example, real-time gaming, stock trading, video conferencing). This Configuration Profile addresses a Kubernetes cluster hardware, software capabilities, and configurations that enable high performance for packet forwarding packets. In this category, you can find workloads such as UPF, vBNG, vCMTS, and vCDN.

- **Regional Data Center Configuration Profile** – The Regional Data Center consists of a management domain with many racks of servers, typically managed and orchestrated by a single instance of resource orchestration. Usage scenarios include services such as content delivery, media, mobile connectivity, and cloud services. This Configuration Profile is tailored exclusively and defined for Media Visual Processing workloads such as CDN Transcoding.
- **On-Premises Edge Configuration Profile** – Small cluster of stationary or mobile server platforms, ranging from one to four servers. Usage scenarios include data collection from sensors, local (edge) processing, and upstream data transmission. Sample locations are hospitals, factory floors, law enforcement, media, cargo transportation, and power utilities. This Configuration Profile recommends a Kubernetes cluster hardware configuration, software capabilities, and specific hardware and software configurations that typically support enterprise edge workloads used in Smart City deployments and ad insertion.
- **On-Premises VSS Configuration Profile** – Small cluster of stationary or mobile server platforms, ranging from one to four servers. Profile recommends a Kubernetes cluster hardware configuration, software capabilities, and specific hardware and software configurations to support Video Structuring Server (VSS).
- **On-Premises AI Box Configuration Profile** – Small cluster of stationary or mobile server platforms, ranging from one to four servers. Profile recommends a Kubernetes cluster hardware configuration, software capabilities, and specific hardware and software configurations to support Intel® Edge AI Box.
- **On-Premises SW Defined Factory Configuration Profile** – Small cluster of stationary or mobile server platforms, ranging from one to four servers. Usage scenarios include data collection from sensors, local (edge) processing, and upstream data transmission. Sample location tuned for factory floors. This Configuration Profile recommends a Kubernetes cluster hardware configuration, software capabilities, and specific hardware and software configurations that typically support industrial workloads used in factory deployments.
- **Access Edge Configuration Profile** – A small cluster designed to support cellular access network deployments, typically in an outside plant in harsh, minimally controlled temperature cabinets. Targeted use cases are 5G Virtual Radio Access Networks (vRAN) and FlexRAN™ 5G solutions that require high throughput, low latency, security, and power management control.

Additional Reference Architecture Configuration Profiles are not location-specific and enable flexible deployments per need:

- **Basic Configuration Profile** – A minimum set of software features where network acceleration is the only concern.
- **Build-Your-Own Configuration Profile** – A complete set of all available software features targeted at developers and deployers that are looking to evaluate, control, and configure all of the software and hardware ingredients and dependencies individually.

The following table links the Quick Start Guides with the Configuration Profiles.

Table 4. Configuration Profiles and Related Quick Start Guides

Configuration Profile	Quick Start Guide
On-Premises Edge	Network and Edge Reference System Architectures - CDN Quick Start Guide
On-Premises VSS	Network and Edge Reference System Architectures - Edge Analytics Video Structuring Server (VSS) Quick Start Guide
On-Premises AI Box	Network and Edge Reference System Architectures - On Premises Edge AI Box Quick Start Guide
On-Premises SW Defined Factory	Network and Edge Reference System Architectures - Industrial Controller Quick Start Guide
Access Edge	▪ Network and Edge Reference System Architectures - vRAN Setup with FlexRAN™ Software Quick Start Guide ▪ Network and Edge Reference System Architectures - 5G vRAN Security Quick Start Guide
Remote Central Office-Forwarding	Network and Edge Reference System Architectures - 5G Core UPF Quick Start Guide
Regional Data Center	Network and Edge Reference System Architectures - Video Production Quick Start Guide
Basic	Network and Edge Reference System Architectures - Single Server Quick Start Guide
Build-Your-Own	Network and Edge Reference System Architectures - Single Server Quick Start Guide

2.3 Reference Architecture Installation Prerequisites

This section helps you get ready to run the Ansible scripts. Before the Ansible playbook can begin, you must identify the required hardware components, ensure hardware connectivity, and complete the initial configuration, for example BIOS setup. This section describes the minimal system prerequisites needed for the Ansible host and Kubernetes control and worker nodes. It also lists the steps required to prepare hosts for successful deployment. Detailed instructions are provided in relative sections, which are referred to in this section. Steps include:

- Hardware BOM selection and setup

- Required BIOS/UEFI configuration, including virtualization and hyper-threading settings
- Network topology requirements – a list of necessary network connections between the nodes
- Installation of software dependencies needed to execute Ansible playbooks
- Generation and distribution of SSH keys that are used for authentication between the Ansible host and Kubernetes cluster target servers

After satisfying these prerequisites, Ansible playbooks for 4th and 5th Gen Intel® Xeon® Scalable processors and Intel® Xeon® D processors can be downloaded directly from the dedicated GitHub page (Container Experience Kits Releases) or cloned using the Git.

2.3.1 Hardware BOM Selection and Setup for Control and Worker Nodes

Before software deployment and configuration, deploy the physical hardware infrastructure for the site. To obtain ideal performance and latency characteristics for a given network location, Intel recommends the hardware BOMs and configurations described in the following sections.

- Hardware Components List for 5th Gen Intel® Xeon® Scalable Processor
- Hardware Components List for 4th Gen Intel® Xeon® Scalable Processor
- Hardware Components List for 4th Gen Intel® Xeon® Scalable Processor with Integrated Intel® vRAN Boost
- Hardware Components List for Intel® Xeon® D Processor
- Hardware Components List for Intel® Core™ Processor
- Hardware Components List for Intel Atom® Processor

2.3.2 BIOS Selection for Control and Worker Nodes

Enter the UEFI or BIOS menu and update the configuration as listed in the tables in Section 3.7 that describe the BIOS selection in detail.

2.3.3 Operating System Selection for Ansible Host and Control and Worker Nodes

The following Linux operating systems are supported for Control and Worker Nodes:

- RHEL for x86_64 version 9.2
- RHEL 9.2 RT
- Rocky Linux 9.2
- Ubuntu 22.04
- Ubuntu 22.04 RT

For all supported distributions, the base operating system installation images are sufficient when built using the "Minimal" option during installation. In addition, the following must be met:

- The Control and Worker Nodes must have network connectivity to the Ansible host.
- All systems must have public internet connectivity.

2.3.4 Network Interface Requirements for Control and Worker Nodes

The following list provides a brief description of different networks and network interfaces needed for deployment.

- Internet network
  - Ansible host accessible
  - Capable of downloading packages from the internet
  - Can be configured for Dynamic Host Configuration Protocol (DHCP) or with static IP address

- Management network and Calico pod network interface (This can be a shared interface with the internet network)
  - Kubernetes control and worker node inter-node communications
  - Calico pod network runs over this network
  - Configured to use a private static address

- Tenant data networks
  - Dedicated networks for traffic
  - Single Root Input/Output Virtualization (SR-IOV) enabled
  - Virtual function (VF) can be DPDK bound in pod

2.4 Ansible Playbook

This section describes how the Ansible playbooks allow for an automated deployment of a fully functional BMRA cluster, including initial system configuration, Kubernetes deployment, and set up of capabilities as described in Section 2.5.

2.4.1 Ansible Playbook Building Blocks

The following components make up the BMRA Ansible playbooks.

Note: Ansible playbooks for 4th and 5th Gen Intel Xeon Scalable processors and Intel® Xeon® D processors are open source and available here.

Configuration Files provide examples of cluster-wide and host-specific configuration options for each of the Configuration Profiles. With minimal changes, these configuration files can be used directly with their corresponding playbooks. The path to these configuration files is:

- `inventory.ini`
- `group_vars/all.yml`
- `host_vars/node1.yml`

For default values in these files, refer to the *BMRA.pdf* file available on GitHub.

2.4.2 Ansible Playbook Phases

Regardless of the selected Configuration Profile, the installation process always consists of three main phases:

1. **Infrastructure Setup** (sub-playbooks in `playbooks/infra/` directory)
   These playbooks modify kernel boot parameters and apply the initial system configuration for the cluster nodes. Depending on the selected Configuration Profile, Infrastructure Setup includes:
   – Generic host OS preparation, e.g., installation of required packages, Linux kernel configuration, proxy and DNS configuration, and modification of SELinux policies and firewall rules.
   – Configuration of the kernel boot parameters according to the user-provided configuration in order to configure CPU isolation, SR-IOV related settings such as IOMMU, hugepages, or explicitly enable/disable Intel P-state technology.
   – Configuration of SR-IOV capable network cards and QAT devices. This includes the creation of virtual functions and binding to appropriate Linux kernel modules.
   – Network Adapter drivers and firmware updates, which help ensure that all of the latest capabilities such as Dynamic Device Personalization (DDP) profiles are enabled.
   – Intel® Speed Select Technology (Intel® SST) configuration, which provides control over base frequency.
   – Installation of DDP profiles, which can increase packet throughput, help reduce latency, and lower CPU usage by offloading packet classification and load balancing to the network adapter.
2. **Kubernetes Setup** (in `playbooks/k8s/` directory)
   This playbook deploys a high availability (HA) Kubernetes (K8s) cluster using Kubespray, which is a project under the Kubernetes community that deploys production-ready Kubernetes clusters. The Multus container network interface (CNI) plugin, which is specifically designed to support multiple networking interfaces in a Kubernetes environment, is deployed by Kubespray along with Calico and Helm. Preferred security practices are used in the default configuration. On top of Kubespray, there is also a container registry instance deployed to store images of various control-plane Kubernetes applications, such as Telemetry Aware Scheduling (TAS), CPU Manager for Kubernetes (CMK), or device plugins.
3. **BMRA System Capabilities Setup** (sub-playbooks in the `playbooks/intel` directory)
   Advanced networking technologies, enhanced platform awareness, and device plugin features are deployed by this playbook using operators or Helm charts as part of the BMRA. The following capabilities are deployed:
   – Device plugins that allow using, for example, SR-IOV, QAT, and GPU devices in workloads running on top of Kubernetes.
   – SR-IOV CNI plugin, Bond CNI plugin, and Userspace CNI plugin, which allow Kubernetes pods to be attached directly to accelerated and highly available hardware and software network interfaces.
   – Native CPU Manager for Kubernetes (replacement for CMK), which performs a variety of operations to enable core pinning and isolation on a container or a thread level.
   – Node Feature Discovery (NFD), which is a Kubernetes add-on to detect and advertise hardware and software capabilities of a platform that can, in turn, be used to facilitate intelligent scheduling of a workload.
   – Telemetry Aware Scheduling (TAS), which allows scheduling workloads based on telemetry data.
   – Full Telemetry Stack consisting of collectd, Kube-Prometheus, Jaeger, OpenTelemetry, and Grafana, which provides cluster and workload monitoring capabilities and acts as a source of metrics that can be used in TAS to orchestrate scheduling decisions.
   – MinIO operator/console, which supports deploying MinIO tenants onto private and public cloud infrastructures ("Hybrid" Cloud).

An overview of the features included in each of the three main phases can be seen in Figure 2. Some of the features have dependencies and other features are mutually exclusive. The actual feature set varies depending on the choice of Configuration Profile, as referenced in Section 2.4.1.

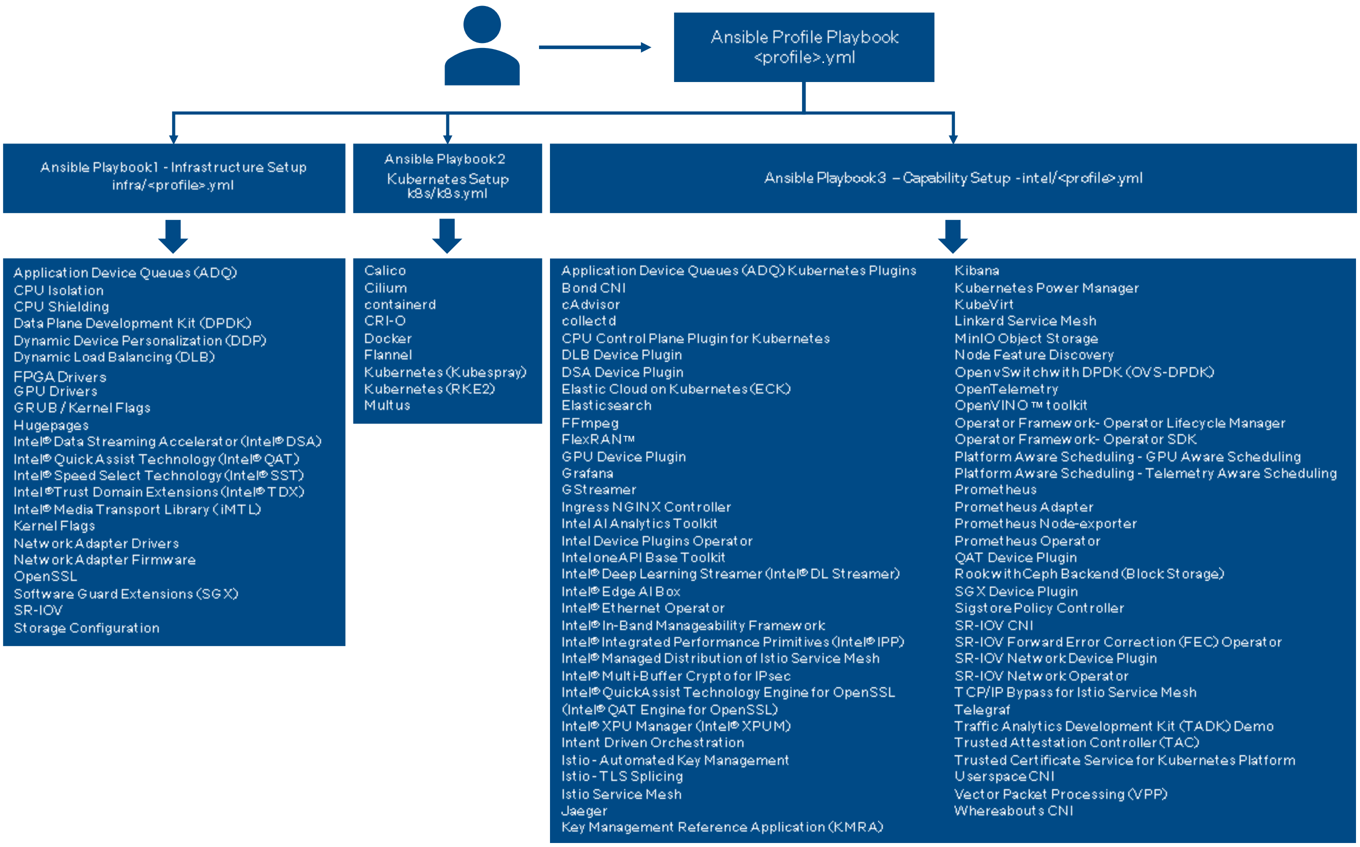
Ansible Profile Playbook
<profile>.yml
Ansible Playbook1 - Infrastructure Setup
infra/<profile>.yml
Ansible Playbook2
Kubernetes Setup
k8s/k8s.yml
Ansible Playbook3 – Capability Setup - intel/<profile>.yml
Application Device Queues (ADQ)
CPU Isolation
CPU Shielding
Data Plane Development Kit (DPDK)
Dynamic Device Personalization (DDP)
Dynamic Load Balancing (DLB)
FPGA Drivers
GPU Drivers
GRUB / Kernel Flags
Hugepages
Intel® Data Streaming Accelerator (Intel® DSA)
Intel® QuickAssist Technology (Intel® QAT)
Intel® Speed Select Technology (Intel® SST)
Intel® Trust Domain Extensions (Intel® TDX)
Intel® Media Transport Library (iMTL)
Kernel Flags
Network Adapter Drivers
Network Adapter Firmware
OpenSSL
Software Guard Extensions (SGX)
SR-IOV
Storage Configuration
Calico
Cilium
containerd
CRI-O
Docker
Flannel
Kubernetes (Kubespray)
Kubernetes (RKE2)
Multus
Application Device Queues (ADQ) Kubernetes Plugins
Bond CNI
cAdvisor
collectd
CPU Control Plane Plugin for Kubernetes
DLB Device Plugin
DSA Device Plugin
Elastic Cloud on Kubernetes (ECK)
Elasticsearch
FFmpeg
FlexRAN™
GPU Device Plugin
Grafana
GStreamer
Ingress NGINX Controller
Intel AI Analytics Toolkit
Intel Device Plugins Operator
Intel oneAPI Base Toolkit
Intel® Deep Learning Streamer (Intel® DL Streamer)
Intel® Edge AI Box
Intel® Ethernet Operator
Intel® In-Band Manageability Framework
Intel® Integrated Performance Primitives (Intel® IPP)
Intel® Managed Distribution of Istio Service Mesh
Intel® Multi-Buffer Crypto for IPsec
Intel® QuickAssist Technology Engine for OpenSSL
(Intel® QAT Engine for OpenSSL)
Intel® XPU Manager (Intel® XPUM)
Intent Driven Orchestration
Istio - Automated Key Management
Istio - TLS Splicing
Istio Service Mesh
Jaeger
Key Management Reference Application (KMRA)
Kibana
Kubernetes Power Manager
KubeVirt
Linkerd Service Mesh
MinIO Object Storage
Node Feature Discovery
Open vSwitch with DPDK (OVS-DPDK)
OpenTelemetry
OpenVINO™ toolkit
Operator Framework - Operator Lifecycle Manager
Operator Framework - Operator SDK
Platform Aware Scheduling - GPU Aware Scheduling
Platform Aware Scheduling - Telemetry Aware Scheduling
Prometheus
Prometheus Adapter
Prometheus Node-exporter
Prometheus Operator
QAT Device Plugin
Rook with Ceph Backend (Block Storage)
SGX Device Plugin
Sigstore Policy Controller
SR-IOV CNI
SR-IOV Forward Error Correction (FEC) Operator
SR-IOV Network Device Plugin
SR-IOV Network Operator
TCP/IP Bypass for Istio Service Mesh
Telegraf
Traffic Analytics Development Kit (TADK) Demo
Trusted Attestation Controller (TAC)
Trusted Certificate Service for Kubernetes Platform
Userspace CNI
Vector Packet Processing (VPP)
Whereabouts CNI

Figure 2. Features Included in Each Ansible Playbook Phase

2.5 Deployment Using Ansible Playbook

This section describes common steps to obtain the BMRA Ansible Playbooks source code, prepare target servers, configure inventory and variable files, and deploy the BMRA Kubernetes cluster.

2.5.1 Prepare Target Servers

For each target server that will act as a control or worker node, you must make sure that it meets the following requirements:

- Python 3 is installed. The following example assumes that the host is running RHEL. Other operating systems may have slightly different installation steps:
  ```
  yum install python3
  ```
- Internet access on all target servers is mandatory. Proxies are supported and can be configured in the Ansible vars.
- BIOS configuration matching the desired profile and use case. For details see Section 3.7 and the specific quick start guide.

2.5.2 Prepare Ansible Host and Configuration Templates

Perform the following steps:

1. Log in to your Ansible host (the one that you will run these Ansible playbooks from).
2. (optional) Configure proxies if necessary:
   - Add proxy configuration to `/etc/environment` (values shown are for example purposes):
     ```
     http_proxy=http://proxy.example.com:1080
     https_proxy=http://proxy.example.com:1080
     ```
   - Update current environment to include proxy configuration from previous step:
     ```
     source /etc/environment
     ```
3. Install packages on Ansible host. The following example assumes that the host is running RHEL. Other operating systems may have slightly different installation steps and some packages may already be present:
   ```
   yum install python3 python3-pip openssh-server git
   pip3 install --upgrade pip
   ```
4. Enable passwordless login between all nodes in the cluster.
   Create authentication ssh-keygen keys on Ansible host:
   ```
   ssh-keygen
   ```

5. SSH is used by the Ansible host to communicate with each target node. Configure the same SSH keys on each machine. Copy your generated public keys to all the nodes from the Ansible host:

   ```
   ssh-copy-id root@<target_server_address>
   ```

6. Clone the source code and change working directory.

   ```
   git clone https://github.com/intel/container-experience-kits/
   cd container-experience-kits
   ```

   Check out the latest version of the playbooks, for example:

   ```
   git checkout v24.01
   ```

   Note: Alternatively go to Container Experience Kits Releases, download the latest release tarball, and unarchive it:

   ```
   wget https://github.com/intel/container-experience-kits/archive/v24.01.tar.gz
   tar xzf v24.01.tar.gz
   cd container-experience-kits-24.01
   ```

7. Decide which Configuration Profile that you want to deploy and export the environmental variable.

   For Kubernetes **Remote Central Office-Forwarding** Configuration Profile deployment:

   ```
   export PROFILE=remote_fp
   ```

   For Kubernetes **Regional Data Center** Configuration Profile deployment:

   ```
   export PROFILE=regional_dc
   ```

   For Kubernetes **On-Premises Edge** Configuration Profile deployment:

   ```
   export PROFILE=on_prem
   ```

   For Kubernetes **On-Premises VSS** Configuration Profile deployment:

   ```
   export PROFILE=on_prem_vss
   ```

   For Kubernetes **On-Premises AI Box** Configuration Profile deployment:

   ```
   export PROFILE=on_prem_aibox
   ```

   For Kubernetes **On-Premises SW Defined Factory** Configuration Profile deployment:

   ```
   export PROFILE=on_prem_sw_defined_factory
   ```

   For Kubernetes **Access Edge** Configuration Profile deployment:

   ```
   export PROFILE=access
   ```

   For Kubernetes **Basic** Configuration Profile deployment:

   ```
   export PROFILE=basic
   ```

   For Kubernetes **Build-Your-Own** Configuration Profile deployment:

   ```
   export PROFILE=build_your_own
   ```

8. Install Python dependencies using one of the following methods:

   - (non-invasive) Virtual environment using `pipenv`:

     ```
     pip3 install pipenv
     pipenv install
     pipenv shell
     ```

   - (non-invasive) Virtual environment using `venv`:

     ```
     python3 -m venv venv
     source venv/bin/activate
     pip3 install -r requirements.txt
     ```

   - (not recommended) System environment:

     ```
     pip3 install -r requirements.txt
     ```

9. Install Ansible collection dependencies:

   ```
   ansible-galaxy install -r collections/requirements.yml
   ```

10. Generate profile. Be aware of the machine's architecture and data plane network before generating profiles.

    ```
    # Supported architectures (ARCH): atom, core, icx, spr, emr
    # Supported data plane network adapters (NIC): fvl, cvl
    make k8s-profile PROFILE=$PROFILE ARCH=spr NIC=cvl
    ```

2.5.3 Update Ansible Inventory File

Perform the following steps:

1. Edit the *inventory.ini* file generated in the previous steps.

   a. In section `[all]`, specify all your target servers. Use their actual hostnames and management IP addresses. Also update `ansible_user` and `ansible_password` to match the SSH configuration of the target servers. If any of the servers are configured with passwordless SSH, the `ansible_password` host variable can be removed.

   b. In sections `[kube_control_plane]` and `[etcd]`, add hostname entries from `[all]` that correspond to target servers that should be used as controller nodes. In `[kube_node]`, add hostname entries from `[all]` for worker nodes.

   ```
   [all]
   ```

```
controller1      ansible_host=10.0.0.1 ip=10.0.0.1 ansible_user=USER ansible_password=XXXX
controller2      ansible_host=10.0.0.2 ip=10.0.0.2 ansible_user=USER ansible_password=XXXX
controller3      ansible_host=10.0.0.3 ip=10.0.0.3 ansible_user=USER ansible_password=XXXX
node1            ansible_host=10.0.0.4 ip=10.0.0.4 ansible_user=USER ansible_password=XXXX
node2            ansible_host=10.0.0.5 ip=10.0.0.5 ansible_user=USER ansible_password=XXXX
localhost        ansible_connection=local ansible_python_interpreter=/usr/bin/python3
[vm_host]
[kube_control_plane]
controller1
controller2
controller3
[etcd]
controller1
controller2
controller3
[kube_node]
node1
node2
[k8s_cluster:children]
kube_control_plane
kube_node

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

2.5.4 Update Ansible Host and Group Variables

Perform the following steps.

1. Create *host_vars/<hostname>.yml* files for all worker nodes, matching their hostnames from the inventory file. The provided *host_vars/node1.yml* file can be used as a template.
2. Edit all *host_vars/<hostname>.yml* and *group_vars/all.yml* files to match your desired configuration. To select Rancher RKE2 as an alternative Kubernetes distribution, you also need set the following variables.

```
# set to rke2 to enable rke2 deployment
kube_provisioner: rke2
# rke2 only supports containerd as container_runtime, default value is docker
container_runtime: containerd
```

Each Configuration Profile uses its own set of variables. Refer to the *BMRA.pdf* file on GitHub for the complete list.

2.5.5 Run Ansible Cluster Deployment Playbook

After the inventory and vars are configured, you can run the provided playbooks from the root directory of the project.

- (Required) Apply required patches for Kubespray:

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

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

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

- (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
```

- Run the main playbook:

```
ansible-playbook -i inventory.ini playbooks/${PROFILE}.yml
```

Pay attention to logs and messages displayed on the screen. Depending on the selected Configuration Profile, network bandwidth, storage speed, and other similar factors, the execution might take between 30-90 minutes.

After the playbook finishes without any failed tasks, you can proceed with the deployment validation described in Section 5.

Note: Additional information can be found in the Ansible project root directory readme.

2.5.6 Run Ansible Cluster Removal Playbook

If the playbook fails or 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
```

After successful removal of Kubernetes components, you can repeat Section 2.5.5.

Note: Any OS and/or hardware configurations (e.g., proxies, drivers, kernel parameters) are not reset by the cleanup playbook.

3 Reference Architecture Hardware Components and BIOS

For all Configuration Profiles, this section provides a menu of all possible hardware components for control node and worker node as well as the BIOS components available.

3.1 Hardware Components List for 4th Gen Intel® Xeon® Scalable Processor

Table 5. Hardware Components for 4th Gen Intel Xeon Scalable Processor

Ingredient	Requirement	Required/ Recommended
4th Gen Intel Xeon Scalable processors	Intel® Xeon® Gold 5418N processor at 2.0 GHz, 24 C/48 T, 165 W, or higher number Intel® Xeon® Gold or Platinum CPU SKU	Required
Memory	DRAM only configuration: 256 GB DRAM (16 x 16 GB DDR5)	Required
Intel® Optane™ Persistent Memory	512 GB (4 x 128 GB Intel® Optane™ persistent memory in 2-1-1 topology)	Recommended
Network Adapter	Intel® Ethernet Network Adapter E810-CQDA2 or E810-XXVDA2	Required
	Intel® Ethernet Controller XXV/ XL 710	
Intel® QAT	Integrated in the processor	
Storage (Boot Drive)	Intel® SATA Solid State Drive D3 S4510 at 480 GB or equivalent boot drive	Required
Storage (Capacity)	Intel® SSD D7-P5510 Series at 3.84 TB or equivalent drive (recommended NUMA aligned)	Recommended
LAN on Motherboard (LOM)	10 Gbps or 25 Gbps port for Preboot Execution Environment (PXE) and Operation, Administration, and Management (OAM)	Required
	1/10 Gbps port for Management Network Adapter	
Additional Plug-in cards	Intel® vRAN Accelerator ACC100 Adapter	Optional
	Intel® Data Center GPU Flex Series	
	Intel® FPGA SmartNIC WSN6050 Platform	

The hardware components listed are for a basic worker or control node.

Some Configuration Profiles may need the worker node to have a CPU upgrade and increase in memory to 512 GB: CPU upgrade: Intel® Xeon® Gold 6438N processor at 1.8 GHz, 32 C/64 T, 205 W

All vRAN configurations require the Intel® vRAN Accelerator ACC100 Adapter and the Intel® Xeon® Gold 6421N processor is recommended.

For configurations that require additional storage, add Kioxia CM6 3.2 TB NVMePCIe4x4 2.5" 15 mm SIE 3DWPD - KCM6XVUL3T20.

3.2 Hardware Components List for 4th Gen Intel® Xeon® Scalable Processor with Integrated Intel® vRAN Boost

Table 6. Hardware Components for 4th Gen Intel Xeon Scalable Processor with Integrated Intel® vRAN Boost

Ingredient	Requirement	Required/ Recommended
4th Gen Intel Xeon Scalable processors	20 or 32 core 4th Gen Intel® Xeon® Scalable processor with integrated Intel® vRAN Boost CPU	Required
Memory	DRAM only configuration: 128 GB DRAM (8 x 16 GB DDR5)	Required
Intel® Optane™ Persistent Memory	512 GB (4 x 128 GB Intel® Optane™ persistent memory in 2-1-1 topology)	Recommended
Network Adapter	Option 1: Intel® Ethernet Network Adapter E810-CQDA2	Required
	Option 2: Intel® Ethernet Network Adapter E810-XXVDA4	
Intel® QAT		Not Required
Storage (Boot Drive)	Intel® SATA Solid State Drive D3 S4510 at 480 GB or equivalent boot drive	Required
Storage (Capacity)	Intel® SSD D7-P5510 Series at 3.84 TB or equivalent drive (recommended NUMA aligned)	Required

Ingredient	Requirement	Required/ Recommended
LAN on Motherboard (LOM)	10 Gbps or 25 Gbps port for Preboot Execution Environment (PXE) and Operation, Administration, and Management (OAM)	Required
	1/10 Gbps port for Management Network Adapter	
Additional Plug-in cards	N/A	

3.3 Hardware Components List for 5th Gen Intel® Xeon® Scalable Processor

Table 7. Hardware Components for 5th Gen Intel Xeon Scalable Processor

Ingredient	Requirement	Required/ Recommended
5th Gen Intel Xeon Scalable processors	5th Gen Intel® Xeon® Scalable processor CPU (XCC, MCC, or EE LCC)	Required
Memory	DRAM only configuration: 256 GB DRAM (16 x 16 GB DDR5)	Required
Network Adapter	Intel® Ethernet Network Adapter E810-CQDA2 or E810-XXVDA2	Required
Intel® QAT	Integrated in the processor	
Storage (Boot Drive)	Intel® SATA Solid State Drive D3 S4510 at 480 GB or equivalent boot drive	Required
Storage (Capacity)	Intel® SSD D7-P5510 Series at 3.84 TB or equivalent drive (recommended NUMA aligned)	Recommended
LAN on Motherboard (LOM)	10 Gbps or 25 Gbps port for Preboot Execution Environment (PXE) and Operation, Administration, and Management (OAM)	Required
	1/10 Gbps port for Management Network Adapter	
Additional Plug-in cards	Intel® Data Center GPU Flex Series	Optional

3.4 Hardware Components List for Intel® Xeon® D Processor

Table 8. Hardware Components for Intel® Xeon® D Processor

Ingredient	Requirement	Required/ Recommended
Intel® Xeon® D processors	Intel® Xeon® D-1700 processor, 4 core LCC, 45 W, or Intel® Xeon® D-1700 processor, 10 core LCC, or Intel Xeon D-2733 NT processor, 8 cores HCC, 80 W	Required
Memory	DRAM only configuration: 32 GB DDR4 2933 MHz	Required
Network Adapter	2 x 10/25 GbE integrated Ethernet ports or Intel® Ethernet Network Adapter E810-CQDA2	Required
Intel® QAT	Intel® QuickAssist Adapter 8960 or 8970 (PCIe*) AIC or equivalent third-party Intel® C620 Series Chipset	Recommended
Storage (Boot Drive)	Intel® SSD 256 GB 2.5" internal SSD/M.2	Required
Additional Plug-in cards	N/A	

The hardware components listed are for a basic worker or control node.

Some Configuration Profiles may need the worker node to have a CPU upgrade and increase memory to 64 GB, storage to 512 GB: CPU upgrade: Intel® Xeon® D-2766NT processor 2.1 GHz, 14 core HCC, 97 W with four integrated Ethernet ports.

3.5 Hardware Components List for Intel® Core™ Processor

Table 9. Hardware Components for 11th Gen Intel® Core™ mobile processor with Intel® Iris® X^e Integrated Graphics

Ingredient	Requirement	Required/ Recommended
Intel® Core™ Processor	11th Gen Intel® Core™ mobile processor with Intel® Iris® X^e Integrated Graphics	Required
Memory	DRAM only configuration: 16 GB DDR4 2933 MHz	Required
Network Adapter	Integrated Ethernet ports	
Storage (Boot Drive)	Intel® SSD 256 GB 2.5" internal SSD/M.2	Required

Ingredient	Requirement	Required/ Recommended
Additional Plug-in cards	N/A	

Table 10. Hardware Components for 12th Gen Intel® Core™ desktop processor

Ingredient	Requirement	Required/ Recommended
Intel® Core™ Processor	12th Gen Intel® Core™ desktop processor (Default i7-12700)	Required
Memory	DRAM only configuration: 16 GB DDR4 2933 MHz	Required
Network Adapter	Integrated Ethernet ports	
Storage (Boot Drive)	Intel® SSD 256 GB 2.5" internal SSD/M.2	Required
Additional Plug-in cards	Intel® Arc™ Discrete Graphics GPU A380	Optional

Table 11. Hardware Components for 12th Gen Intel® Core™ mobile processor with Intel® Iris® X^e Integrated Graphics

Ingredient	Requirement	Required/ Recommended
Intel® Core™ Processor	12th Gen Intel® Core™ mobile processor with Intel® Iris® X^e Integrated Graphics	Required
Memory	DRAM only configuration: 16 GB DDR4 2933 MHz	Required
Network Adapter	Integrated Ethernet ports	
Storage (Boot Drive)	Intel® SSD 256 GB 2.5" internal SSD/M.2	Required
Additional Plug-in cards	N/A	

Table 12. Hardware Components for 12th Gen Intel® Core™ processor for IoT Edge with Intel® Iris® X^e Integrated Graphics

Ingredient	Requirement	Required/ Recommended
Intel® Core™ Processor	12th Gen Intel® Core™ processor for IoT Edge with Intel® Iris® X^e Integrated Graphics	Required
Memory	DRAM only configuration: 16 GB DDR4 2933 MHz	Required
Network Adapter	Integrated Ethernet ports	
Storage (Boot Drive)	Intel® SSD 256 GB 2.5" internal SSD/M.2	Required
Additional Plug-in cards	N/A	

Table 13. Hardware Components for 13th Gen Intel® Core™ mobile processor with Intel® Iris® X^e Integrated Graphics

Ingredient	Requirement	Required/ Recommended
Intel® Core™ Processor	13th Gen Intel® Core™ mobile processor with Intel® Iris® X^e Integrated Graphics	Required
Memory	DRAM only configuration: 16 GB DDR4 2933 MHz	Required
Network Adapter	Integrated Ethernet ports	
Storage (Boot Drive)	Intel® SSD 256 GB 2.5" internal SSD/M.2	Required
Additional Plug-in cards	N/A	

Table 14. Hardware Components for Intel® Core™ Ultra Processors for Edge Deployments (includes the Intel® Arc and Intel® AI Boost)

Ingredient	Requirement	Required/ Recommended
Intel® Core™ Processor	Intel® Core™ Ultra Processor with integrated Intel® Arc™ and Intel® AI Boost	Required
Memory	DRAM only configuration: 16 GB DDR5 5600 MHz	Required
Network Adapter	Integrated Ethernet ports	
Storage (Boot Drive)	Intel® M.2 NVME 256 GB SSD	Required
Additional Plug-in cards	N/A	

3.6 Hardware Components List for Intel Atom® Processor

Table 15. Hardware Components for Intel Atom® Processor

Ingredient	Requirement	Required/ Recommended
Intel Atom® processor	Intel Atom® processor x6000 series (Default x6425RE)	Required
Memory	DRAM only configuration: 16 GB DDR4 2933 MHz	Required
Network Adapter	Integrated Ethernet ports	
Storage (Boot Drive)	Intel® SSD 256 GB 2.5" internal SSD/M.2	Required
Additional Plug-in cards	N/A	

3.7 Platform BIOS

This section provides different BIOS Profiles required for the BMRA Configuration Profiles. For more information about BIOS settings, visit the Intel BIOS Setup Utility User Guide.

3.7.1 4th and 5th Gen Intel® Xeon® Scalable Processor Platform BIOS

Table 16. Platform BIOS Settings for 4th and 5th Gen Intel® Xeon® Scalable Processor

Menu (Advanced)	Path to BIOS Setting	BIOS Setting	Low Latency	Max Performance with Turbo	Energy Balance Turbo
Socket Configuration	Processor Configuration	Hyper-Threading	Enable	Enable	Enable
		X2APIC	Enable	Enable	Enable
		VMX	Enable	Enable	Enable
		Homeless Prefetch	Enable	Disable (default)	Disable (default)
		LLC Prefetch	Disable	Enable	Enable
		SNC	Disable	Disable	Disable
		Uncore RAPL	Disable	Disable	Enable
		Uncore frequency scaling	Disable	Disable	Enable
		Uncore frequency	1.8 GHz (hex 0x12)	1.6 MHz (hex 0x10)	800 MHz to 2.5 GHz
Power Configuration	CPU P-state Control	EIST PSD Function	HW_ALL	HW_ALL	HW_ALL
		Boot Performance Mode	Max. Performance	Max. Performance	Max. Performance
		AVX License Pre-Grant	Enable	Disable	Disable
		AVX ICCP Pre Grant Level	Level 5	NA	NA
		AVX P1 (Config TDP)	Level 2	Nominal (default)	Nominal
		Energy Efficient Turbo	Disable	Disable	Enable

Menu (Advanced)	Path to BIOS Setting	BIOS Setting	Low Latency	Max Performance with Turbo	Energy Balance Turbo
		GPSS timer	0 µs	0 µs	0 µs
		Turbo	Enable	Enable	Enable
		Intel® SpeedStep® Technology	Enable	Enable	Enable
	Frequency Prioritization	RAPL Prioritization	Disable	Disable	Disable
	Common Ref Code	UMA-Based Clustering	Quadrant	Quadrant	Quadrant
	Hardware PM State Control	Hardware P-states	Native with no Legacy Support	Native with no Legacy Support	Native with no Legacy Support
		EPP enable	Disable	Disable	Disable
	CPU C-state Control	Enable Monitor Mwait	Enable	Enable	Enable
		CPU C1 Auto Demotion	Disable	Disable	Disable
		CPU C1 Auto unDemotion	Disable	Disable	Disable
		Processor C6 or CPU C6 Report	Enable	Enable	Enable
		Enhanced Halt State (C1E)	Enable (per Core Level)	Enable	Enable
		OS ACPI Cx	ACPI C2	ACPI C2	ACPI C2
	Energy Performance Bias	Power Performance Tuning	OS Control EPB	OS Controls EPB	OS Controls EPB
		Workload Configuration	I/O Sensitive	I/O Sensitive	Balanced
	Package C-state Control	Package C-state	C6 Retention	C0/C1 State	C0/C1 State
		Dynamic L1	Enable	Disable	Disable
Memory Configuration		Memory Configuration	8-way interleave	8-way interleave	8-way interleave
		Enforce POR / Memory Patrol Scrub	Enable/Disable	Enable/Enable	Enable/Enable
		Memory DIMM Refresh Rate	1x	1x	2x
Platform Configuration	Miscellaneous Configuration	Serial Debug Message Level	Minimum	Minimum	Minimum
	PCI Express* Configuration	PCIe* ASPM	Disable	Enable	Enable
		ECRC generation and checking	Disable	Enable	Enable
Server Management		Resume on AC Power Loss	Power On	Power On	Power On
System Acoustic and Performance Configuration		Set Fan Profile	Performance	Acoustic	Acoustic

3.7.2 Intel® Xeon® D Processor Platform BIOS

Table 17. Platform BIOS Settings for Intel® Xeon® D Processor

Menu (Advanced)	Path to BIOS Setting	BIOS Setting	Energy Balance	Max Performance	Deterministic
Power Configuration	Power and Performance	CPU Power and Performance Policy	Balanced Performance	Performance	Performance
		Workload Configuration	I/O sensitive	I/O sensitive	I/O sensitive
		Turbo	Disabled	Enabled	Disabled
	CPU P-state control	Enhanced Intel SpeedStep® Technology	Enabled	Enabled	Disabled
		GPSS timer	500 µs	0 µs	0 µs

Menu (Advanced)	Path to BIOS Setting	BIOS Setting	Energy Balance	Max Performance	Deterministic
	Hardware P-states	Hardware P-states	Native Mode with no legacy Support	Disabled	Disabled
	CPU C-state Control	Package C-state	C6 Retention	C6 Retention	C0/C1 State
		C1E	Enabled	Enabled	Disabled
		Processor C6	Enabled	Enabled	Disabled
	Uncore Power Management	Uncore Frequency scaling	Enabled	Disabled	Disabled
		Performance P-limit	Enabled	Disabled	Disabled
Memory Configuration	Memory Configuration	IMC Interleaving	2-way interleave	2-way interleave	2-way interleave
Thermal Configuration	System Acoustic and Performance Configuration	Set Fan Profile	Acoustic	Performance	Performance
GPU	GPU Fz	Lock 900 MHz	Optional	Optional	Optional

3.7.3 Intel® Speed Select Technology BIOS Settings for Intel® Xeon® Processors

Use the following table to configure the BIOS settings to use Intel® Speed Select Technology – Base Frequency (Intel® SST-BF), Intel® Speed Select Technology – Turbo Frequency (Intel® SST-TF), and Intel® Speed Select Technology – Performance Profile (Intel® SST-PP) in 4th and 5th Gen Intel Xeon Scalable processor systems.

Table 18. BIOS Settings to Enable Intel SST-BF, Intel SST-TF, and Intel SST-PP

BIOS Setting	Status
Hardware PM State Control	
Scalability	Disable
Hardware PM Interrupt	Disable
CPU P-state	
Dynamic SST-PP	Enable
Speed Step (P-states)	Enable
Activate SST-BF	Enable
Configure SST-BF	Enable
EIST PSD Function	HW_All
Turbo	Enable
Energy Efficient Turbo	Enable
Boot Performance	Max
Freq: Prioritization AC	
SST-CP	Enable

3.7.4 SGX BIOS Settings for Xeon Processors

In BIOS, the configuration paths might be slightly different, depending on platform, but the key settings are as follows and must be performed in order.

Table 19. BIOS Settings to Enable Intel® SGX on 4th and 5th Gen Intel Xeon Scalable Processor

BIOS Setting	Status
Advanced > Processor Configuration > Total Memory Encryption (TME)	Enable
Advanced > Memory Configuration > Memory RAS and Performance Configuration > UMA-Based Clustering	Disable (All2All)
Advanced > Processor Configuration > SW Guard Extensions (SGX)	Enable
Advanced > Processor Configuration > Enable/Disable SGX Auto MP Registration Agent	Enable

3.7.5 TDX with SGX BIOS Settings for Intel® Xeon® Processors

Table 20. BIOS Settings to Enable Intel® TDX and Intel® SGX on 5th Gen Intel Xeon Scalable Processor

BIOS Setting	Status
Advanced > Processor Configuration > Total Memory Encryption (TME)	Enable
Advanced > Memory Configuration > Memory RAS and Performance Configuration > UMA-Based Clustering	Disable (All2All)
Advanced > Processor Configuration > SW Guard Extensions (SGX)	Enable
Advanced > Processor Configuration > Enable/Disable SGX Auto MP Registration Agent	Enable
Advanced > Processor Configuration > Trust Domain Extensions (TDX)	Enable
Advanced > Processor Configuration > TDX Key Split	Non-zero Value

3.7.6 Intel® Core™ Processor Platform BIOS

Use the default BIOS settings.

3.7.7 Intel Atom® Processor Platform BIOS

Use the default BIOS settings.

4 Reference Architecture Software Components

4.1 Software Components Supported

Table 21 lists the software components automatically deployed per Configuration Profile in a BMRA and their sources.

Table 21. Software Components

Software Function	Software Component	Location
OS	Ubuntu 22.04	https://www.ubuntu.com
	Ubuntu 22.04 RT	
OS	RHEL 9.2	https://www.redhat.com/
OS	RHEL 9.2 RT	https://www.redhat.com/
OS	Rocky 9.2	https://rockylinux.org/
Data Plane Development Kit (DPDK)	v23.11	https://core.dpdk.org/download/
Open vSwitch with DPDK	v3.2.1	https://github.com/openvswitch/ovs
Vector Packet Processing (VPP)	2310	https://packagecloud.io/fdio/
Telegraf	1.3.0	https://github.com/intel/observability-telegraf
Collectd - image	v1.0	https://github.com/intel/observability-collectd/releases/
Collectd - exporter image	v0.5.0	https://github.com/prometheus/collectd_exporter/releases
Grafana	10.2.2	https://www.grafana.com/
Prometheus	2.48.0	https://quay.io/repository/prometheus/prometheus?tab=tags
Container Registry	2.8.3	https://github.com/distribution/distribution/tags
Nginx-image	1.25.2-alpine (kubespray) 1.25.3 (registry)	https://github.com/docker-library/docs/tree/master/nginx
Ansible	Ansible: 8.6.1	https://www.ansible.com/
	Ansible-core: 2.15.8	
BMRA Ansible Playbook	v24.01	https://github.com/intel/container-experience-kits
Python	Python 3.9+	https://www.python.org/
Kubespray	6b1188e commit	https://github.com/kubernetes-sigs/kubespray
Docker	20.10.20	https://www.docker.com/
Helm	v3.13.1	https://github.com/helm/helm
containerd	1.7.8	https://github.com/containerd/containerd/tags
CRI-O	1.28.1	https://github.com/cri-o/cri-o/tags
crictl	1.28.0	https://github.com/kubernetes-sigs/cri-tools/releases
Container orchestration engine	Kubernetes v1.28.3	https://github.com/kubernetes/kubernetes
Rancher	v1.28.3+rke2r1	https://github.com/rancher/rke2/releases
etcd	v3.5.9	https://github.com/etcd-io/etcd/tags
cri-dockerd	0.3.4	https://github.com/Mirantis/cri-dockerd/releases
runc	1.1.9	https://github.com/opencontainers/runc/releases
Platform Aware Scheduling (TAS)	TAS 0.6.0	https://github.com/intel/platform-aware-scheduling
Platform Aware Scheduling (GAS)	GAS 0.5.4	https://github.com/intel/platform-aware-scheduling
Prometheus node-exporter	1.7.0	https://quay.io/repository/prometheus/node-exporter?tab=tags
Prometheus Operator	0.69.1	https://quay.io/repository/prometheus-operator/prometheus-operator?tab=tags
K8s kube-rbac-proxy	0.15.0	https://github.com/brancz/kube-rbac-proxy/releases

Software Function	Software Component	Location
Node Feature Discovery	0.14.3-minimal	https://github.com/kubernetes-sigs/node-feature-discovery
Multus CNI	3.9.3	https://github.com/k8snetworkplumbingwg/multus-cni/tags
Calico	v3.26.3	https://github.com/projectcalico/calico/tags
Cilium	v1.13.4	https://github.com/cilium/cilium/tags
flannel	v0.22.0	https://github.com/flannel-io/flannel/tags
SR-IOV CNI	v2.7.0 (commit 75f2f5e)	https://github.com/k8snetworkplumbingwg/sriov-cni/releases
SR-IOV network device plugin	3.6.2	https://github.com/k8snetworkplumbingwg/sriov-network-device-plugin/releases/
SR-IOV Network Operator	1.2.0	https://github.com/k8snetworkplumbingwg/sriov-network-operator
InfiniBand SR-IOV CNI plugin	v1.0.3	https://github.com/k8snetworkplumbingwg/ib-sriov-cni/tags
Whereabouts CNI	05cc22a	https://github.com/k8snetworkplumbingwg/helm-charts.git
Device Plugins Operator	0.28.0	https://github.com/intel/intel-device-plugins-for-kubernetes
QAT device plugin	0.28.0	https://github.com/intel/intel-device-plugins-for-kubernetes
GPU device plugin	0.28.0	https://github.com/intel/intel-device-plugins-for-kubernetes
Intel® SGX device plugin	0.28.0	https://github.com/intel/intel-device-plugins-for-kubernetes
Intel® DLB device plugin	0.28.0	https://github.com/intel/intel-device-plugins-for-kubernetes
Intel® DSA device plugin	0.28.0	https://github.com/intel/intel-device-plugins-for-kubernetes
Userspace CNI	1.3	https://github.com/intel/userspace-cni-network-plugin
Bond CNI plugin	408b549	https://github.com/k8snetworkplumbingwg/bond-cni
Intel® Ethernet Drivers	i40e v2.23.17	https://sourceforge.net/projects/e1000/files/i40e%20stable/
	ice v1.12.7	https://sourceforge.net/projects/e1000/files/ice%20stable/
	iavf v4.8.2	https://sourceforge.net/projects/e1000/files/iavf%20stable/
Non-Volatile Memory (NVM) Update Utility for Intel® Ethernet Network Adapter 700 Series	9.30	https://www.intel.com/content/www/us/en/download/18190/non-volatile-memory-nvm-update-utility-for-intel-ethernet-network-adapter-700-series.html
Non-Volatile Memory (NVM) Update Utility for Intel® Ethernet Network Adapters E810 Series	4.30	https://www.intel.com/content/www/us/en/download/19626/non-volatile-memory-nvm-update-utility-for-intel-ethernet-network-adapters-e810-series-linux.html
DDP Profiles	Intel® Ethernet 800 Series Dynamic Device Personalization (DDP) for Telecommunication (Comms) Package 1.3.45.0	https://www.intel.com/content/www/us/en/download/19660/intel-ethernet-800-series-telecommunication-comms-dynamic-device-personalization-ddp-package.html
Intel® Ethernet Operator	23.08	https://github.com/intel/intel-ethernet-operator
Operator SDK	1.32.0	https://github.com/operator-framework/operator-sdk
Intel® Ethernet UFT	22.11	https://github.com/intel/UFT.git
Intel® QAT Drivers	QAT20.L.1.0.50-00003	https://www.intel.com/content/www/us/en/download/765501/intel-quickassist-technology-driver-for-linux-hw-version-2-0.html
Intel® QAT Driver Card	QAT.L.4.23.0-00001	https://www.intel.com/content/www/us/en/download/19734/intel-quickassist-technology-driver-for-linux-hw-version-1-7.html?wapkw=qat%20driver
Intel QATLib	23.08.0	https://github.com/intel/qatlib/tags
OpenSSL	openssl-3.1.4	https://github.com/openssl/openssl https://www.openssl.org/source/
OpenSSL QAT Engine	1.4.0	https://github.com/intel/QAT_Engine
Intel ipsec-mb	1.4	https://github.com/intel/intel-ipsec-mb

Software Function	Software Component	Location
Intel® SGX DCAP Drivers	1.19.100.3	https://download.01.org/intel-sgx/sgx-dcap/
Intel® SGX SDK (RHEL)	2.22.100.3	https://download.01.org/intel-sgx/sgx-dcap/1.19/linux/distro/rhel9.2-server/
Intel® SGX SDK (Ubuntu)	2.22.100.3	https://download.01.org/intel-sgx/sgx-dcap/1.19/linux/distro/ubuntu22.04-server/
Intel® KMRA	2.4	https://www.intel.com/content/www/us/en/developer/topic-technology/open/key-management-reference-application/overview.htm
Intel® KMRA AppHSM	2.4	https://hub.docker.com/r/intel/apphsm
Intel® KMRA CTK	2.4	https://hub.docker.com/r/intel/ctk_loadkey
Intel® KMRA PCCS	2.4	https://hub.docker.com/r/intel/pccs
Istio* operator	1.20.1	https://github.com/istio/istio
Intel Istio operator	1.19.0-intel.0	https://github.com/intel/istio
istio-intel/istioctl	1.19.0-intel.0	https://hub.docker.com/r/intel/istioctl/
istio-intel/pilot	1.19.0-intel.0	https://hub.docker.com/r/intel/pilot/
istio-intel/proxyv2	1.19.0-intel.0	https://hub.docker.com/r/intel/proxyv2/
Trusted Attestation Controller	0.4.0	https://github.com/intel/trusted-attestation-controller
Trusted Certificate Service	0.5.0	https://github.com/intel/trusted-certificate-issuer
MinIO Operator	v4.5.8	https://github.com/minio/operator/tree/master/helm/operator
MinIO console	0.22.5	https://github.com/minio/console
local volume provisioner	2.5.0	https://github.com/kubernetes-sigs/sig-storage-local-static-provisioner
Kubernetes Power Manager	2.3.1	https://github.com/intel/kubernetes-power-manager
FEC Operator	0256562 commit	https://github.com/smart-edge-open/sriov-fec-operator
Operator Lifecycle Manager	0.26.0	https://github.com/operator-framework/operator-lifecycle-manager
Operator Package Manager	1.32.0	https://github.com/operator-framework/operator-registry/releases/
FlexRAN™ host software	23.11	NDA
Intel pf-bb-config	23.11	NDA
FlexRAN™ POD software (ICX)	23.07	https://hub.docker.com/r/intel/flexran_vdu
FlexRAN™ POD software (SPR-EE)	23.07	https://hub.docker.com/r/intel/flexran_l1_spree
OpenTelemetry	0.43.0	https://github.com/open-telemetry/opentelemetry-operator
Jaeger	1.51.0	https://github.com/jaegertracing/jaeger-operator
cAdvisor	0.47.2	https://github.com/google/cadvisor/releases
Linkerd	2.14.6	https://helm.linkerd.io/
TADK	23.03.0	https://hub.docker.com/r/intel/tadk-waf
ADQ-K8s-plugin	22.06-1	https://github.com/intel/adq-k8s-plugins
Intel OneAPI Base Toolkit	2023.2.0	https://www.intel.com/content/www/us/en/developer/tools/oneapi/ai-analytics-toolkit-download.html
Intel OneAPI AI analytics kit	2023.2.0	https://www.intel.com/content/www/us/en/developer/tools/oneapi/ai-analytics-toolkit-download.html
Go Lang	1.21.4	https://go.dev/dl/
Intel CPU Control Plane Plugin	0.1.2	https://github.com/intel/cpu-control-plane-plugin-for-kubernetes
Rook Ceph	v1.12.9	https://github.com/rook/rook.git
Multus-service	sha256:3d825327b9851d6045448d7db5323689d1eb4ae813eeac26ed8fcf5ee8d194fb	https://github.com/k8snetworkplumbingwg/multus-service/pkgs/container/multus-service

Software Function	Software Component	Location
Libvirt	9.9.0	https://github.com/libvirt/libvirt
Intel® DL Streamer	2023.0	https://github.com/dlstreamer/dlstreamer
OpenVINO™	2023.1.0	https://github.com/openvinotoolkit/openvino
cartwheel-ffmpeg	2023q3	https://github.com/intel/cartwheel-ffmpeg
gpu stack	20231219	https://dgpu-docs.intel.com/releases
opencv	4.8.0	https://github.com/opencv/opencv
NetConfServer	libyang v2.1.111	https://github.com/CESNET/libyang
	sysrepo v2.2.105	https://github.com/sysrepo/sysrepo
	libnetconf2 v2.1.37	https://github.com/CESNET/libnetconf2
	netopeer2 v2.1.71	https://github.com/CESNET/netopeer2
Intel XPU Manager	v1.2.13	https://github.com/intel/xpumanager
Intel Media Transport Library	v23.08	https://github.com/OpenVisualCloud/Media-Transport-Library/releases
TDX	2023ww41	https://github.com/intel/tdx-tools
Intent-Driven Orchestration (IDO)	v0.2.0	https://github.com/intel/intent-driven-orchestration
Calico VPP Dataplane	3.26.0	https://github.com/projectcalico/vpp-dataplane/releases
Intel® In-Band Manageability Framework	4.1.4	https://github.com/intel/intel-inb-manageability
ECK (Elasticsearch on Kubernetes)	2.10.0	https://github.com/elastic/cloud-on-k8s/releases
Elasticsearch	8.11.3	https://github.com/elastic/elasticsearch/releases
ingress-nginx	4.8.3	https://github.com/kubernetes/ingress-nginx/releases
KubeVirt	v1.1.0	https://github.com/kubevirt/kubevirt/releases

5 Post Deployment Verification Guidelines

This section describes a set of processes that you can use to verify the components deployed by the scripts. The processes are not Configuration Profile-specific but relate to individual components that may not be available in all profiles.

Many verification guidelines and output examples can be found on GitHub, as listed in Table 22, and others are described after the table.

Table 22. Links to Verification Guidelines on GitHub

Verification Step
Check the Kubernetes Cluster
Check Intel SST on Intel Xeon Scalable Processors
Check DDP Profiles
Check Node Feature Discovery
Check Topology Manager
Check SR-IOV Network Operator
Check SR-IOV Device Plugin
Check QAT Device Plugin
Check SGX Device Plugin
Check DSA Device Plugin
Check GPU Device Plugin
Check Multus CNI Plugin
Check SR-IOV CNI Plugin
Check Userspace CNI Plugin
Check Bond CNI Plugin
Check Telemetry Aware Scheduling
Check Intel® Server GPU Device and Driver
Check Intel QAT Engine with OpenSSL
Check MinIO Operator/Console and Tenant
Check Intel Power Manager (Balance Performance Power-Profile & Sample Power-Pods)

5.1 Check Grafana Telemetry Visualization

BMRA deploys Grafana for telemetry visualization. It is available on every cluster node on port 30000. Due to security reasons, this port is not exposed outside the cluster by default. Default credentials are `admin`/`admin` and you should change the default password after first login.

The Grafana TLS certificate is signed by the cluster certificate authority (CA) and it is available in `/etc/kubernetes/ssl/ca.crt`

Visit Grafana at `https://<node-ip>:30000/`

BMRA comes with a set of dashboards from the kube-prometheus project (kube-prometheus). Dashboards are available in the Dashboards > Manage menu.

5.2 Check Key Management Infrastructure with Intel® SGX

To verify the Key Management infrastructure with SGX and use the private keys provisioned to Intel SGX enclaves, see Section 6.1 for step-by-step instructions to set up and run the NGINX workload.

6 Workloads and Application Examples

This section provides examples of how to provision and deploy example applications or workloads.

6.1 Enabling Key Management NGINX Applications

KMRA source code and Dockerfiles: Key Management Reference Application

KMRA docker images on Docker Hub:

- AppHSM: https: //hub.docker.com/r/intel/apphsm
- ctk_loadkey: https://hub.docker.com/r/intel/ctk_loadkey
- PCCS: https://hub.docker.com/r/intel/pccs

KMRA Helm charts are in `/roles/kmra_install/charts`.

Steps to deploy the full KMRA NGINX demo:

1. Generate a new PCCS primary API key and update the `kmra.pccs.api_key` variable in `group_vars/all.yml` (go to Intel® Provisioning Certification Service for ECDSA Attestation and subscribe).
2. Ensure that the `kmra_deploy_demo_workload` variable in the `group_vars/all.yml` is set to `true.`
3. Deploy the `on_prem` or `remote_fp` profile to set up KMRA demo with NGINX. The `kmra` variable must be set to `on` in `profiles/profiles.yml`.

6.2 Enabling Trusted Certificate Service

Trusted Certificate Service (TCS) is a Kubernetes certificate signing solution that uses the security capabilities provided by Intel® SGX. The signing key is stored and used inside the SGX enclaves and is never stored in clear anywhere in the system. TCS is implemented as a cert-manager external issuer by supporting both cert-manager and Kubernetes certificate signing APIs.

To enable TCS on BMRA, follow the guide available at Trusted Certificate Issuer.

6.2.1 Istio Custom CA Integration Using Kubernetes CSR

Istio supports integrating custom certificate authority (CA) using Kubernetes CSR as an experimental feature.

Detailed example steps described in the Istio integration with custom CA using Kubernetes CSR document show how to provision Istio workload certificates using an Issuer provided by the Trusted Certificate Service (TCS).

Note: Due to misconfiguration of the Istio Demo application, you might need to disable hugepages temporarily to avoid the demo app becoming stuck in the `CrashLoopBackOff` state. To disable hugepages, execute the following command on the worker node:

```
echo 0 > /proc/sys/vm/nr_hugepages
```

6.2.2 Remote Attestation and Manual Key Management

TCS supports SGX remote attestation and the sample key management reference application.

All required steps are described in the Remote attestation and key management (manual) document.

6.3 Service Mesh Automated Remote Attestation and Key Management with KMRA, TCS, and TCA

Remote attestation is an advanced feature that allows an entity to gain the relying party's trust. Remote attestation gives the relying party increased confidence that the software is running inside an SGX enclave. The attestation results include the identity of the software being attested and an assessment of possible software tampering.

Key management enables external key management systems to deliver the certificates and keys via more secure mechanisms into the SGX enclave. To enable the automated key management feature, KMRA AppHSM, and KMRA PCCS applications must be enabled and configured as well as Trusted Certificate Service (TCS) and Trusted Certificate Attestation (TCA). BMRA tries to install all dependencies and configure the host with reasonable defaults.

KMRA application settings are collected under the `kmra` variable in the `group_vars/all.yml` file and all default values are available for reference in the `roles/kmra_install/defaults/main.yml` file. If you need to overwrite any default value, redefine it in the `group_vars/all.yml` file while keeping the variable structure.

In general, TCS does not require specific configuration. Default values used for TCS deployment are collected in the `roles/tcs_install/vars/main.yml` file and can be redefined in the `group_vars/all.yml` file.

TCA depends on settings of KMRA AppHSM, which should match. Refer to the default values, which can be found in the `roles/tca_install/vars/main.yml file.` Default values can be redefined in the `group_vars/all.yml file.`

Service mesh default settings can be found in the `roles/service_mesh_install/vars/main.yml file.`

For detailed documentation on components involved in this feature, refer to:

- KMRA: Key Management Reference Application
- TCS: Trusted Certificate Issuer
- TCA: Trusted Attestation Controller

6.4 Istio TLS Splicing

To configure Istio with TLS splicing, first enable it in the `group_vars/all.yml` file.

```
service_mesh:
  enabled:true
  tls_splicing:
    enabled: true
```

The config creates an ingress gateway to act as a forward proxy and registers virtual service rule and external service entry to implement TLS passthrough for external service.

A client outside the mesh can use the cluster ingress gateway to access external services with TLS splicing.

```
export INGRESS_PORT=$(kubectl -n istio-system get service istio-ingressgateway -o
jsonpath='{.spec.ports[?(@.name=="http2")].nodePort}')
export SECURE_INGRESS_PORT=$(kubectl -n istio-system get service istio-ingressgateway -o
jsonpath='{.spec.ports[?(@.name=="https")].nodePort}')
export TCP_INGRESS_PORT=$(kubectl -n istio-system get service istio-ingressgateway -o
jsonpath='{.spec.ports[?(@.name=="tcp")].nodePort}')
export INGRESS_HOST=$(kubectl get po -l istio=ingressgateway -n istio-system -o
jsonpath='{.items[0].status.hostIP}')

curl -s -v  --resolve www.example.com:$SECURE_INGRESS_PORT:$INGRESS_HOST
https://www.example.com:$SECURE_INGRESS_PORT
```

6.5 Web Application Firewall Using Traffic Analytics Development Kit

The functionality of the Web Application Firewall (WAF) running in the cluster can be tested from the command line. Start by getting the IP and port of the firewall:

```
# export NODE_PORT=$(kubectl get --namespace modsec-tadk -o
jsonpath="{.spec.ports[0].nodePort}" services tadk-intel-tadkchart)

# export NODE_IP=$(kubectl get nodes --namespace modsec-tadk -o
jsonpath="{.items[0].status.addresses[0].address}")
```

Note: If the `kube_proxy_nodeport_addresses_cidr` option in `group_vars` has not been commented, the nodeport (`NODE_IP`) will not be available externally. In that, case, replace `NODE_IP` with `localhost`.

Start by verifying that the NGINX server can be reached:

```
## If nodeports are not available externally (default):
# curl http://localhost:$NODE_PORT

## If nodeports are available externally
# curl http://$NODE_IP:$NODE_PORT
```

The output should be the default "Welcome to nginx" webpage.

Now try sending a message with sample credentials to the firewall:

```
## If nodeports are not available externally (default):
# curl -d "username=admin&password=unknown' or '1'='1" "localhost:$NODE_PORT"

## If nodeports are available externally
# curl -d "username=admin&password=unknown' or '1'='1" "$NODE_IP:$NODE_PORT"
```

The resulting error code should be "403" (Forbidden), showing the firewall has blocked the request.

Appendix A Reference System Release Notes

This section lists the notable changes from the previous releases, including new features, bug fixes, and known issues for BMRA, VMRA, and Cloud RA.[4]

A.1 Reference System 24.01 Release Notes

New Components/Features:

- Support 5th Gen Intel® Xeon® Scalable processor for the Edge/IoT
- Support Intel® Edge AI Box 4.0 on 12th Gen Intel® Core™ desktop processors, 12th Gen Intel® Core™ processor for IoT Edge, and Intel® Core™ Ultra Processor for Edge Deployments
- Support Intel® In-Band Manageability framework software to enable an administrator to perform critical device management operations for Intel® Edge AI box
- Support Intel® Workload Services Framework version of the Istio* Envoy* workload
- Support VMs provision via KubeVirt* (v1.1.0)
- Support external access to services via Ingress controller for Kubernetes using NGINX* as a reverse proxy and load balancer (ngress-nginx v4.8.3)
- Support automated detection and configuration of SR-IOV and Intel® QuickAssist Technology (Intel® QAT) devices for VMRA

Updates/Changes:

- Versions upgraded for the vast majority of RA components (See Section 4.1 for all supported versions)
- Notable updates:
  - FlexRAN™ to v23.11 [under NDA]
  - Kubernetes* to v1.28.3
  - Rancher to 2.7.9
  - Data Plane Development Kit (DPDK) to v23.11
  - Open vSwitch with DPDK to 3.2.1
  - Intel® Multi-Buffer Crypto for IPsec library to v1.5
  - Intel® QuickAssist Technology Engine for OpenSSL* (Intel® QAT Engine for OpenSSL*) to v1.4
  - OpenSSL* to 3.1.4
  - Intel® Power Manager to v2.3.1
  - SR-IOV FEC Operator to v2.8
  - OpenVINO™ v2023.1
  - FFmpeg to v2023q3
  - Intel® Deep Learning Streamer (Intel® DL Streamer) 2023.0
  - OpenCV 4.8.0
  - VPP to 23.10
  - Intel® Dynamic Load Balancer (Intel® DLB) software release v8.7.0
  - Intel® Data Streaming Accelerator (Intel® DSA) software release v4.14
  - Istio 1.20.1
  - Intel® Node Feature Discovery (NFD) to 0.14.3
  - Linux overlay kernel 6.3.0-x
  - Intel® Core™ Ultra processor GPU driver and X11/Wayland* UI framework
  - Intel® AI Boost driver for Intel® Core™ Ultra processor [under NDA]
  - Intel® Core™ Ultra processor audio firmware and topology [under NDA]

Updates/Changes made for the Reference System 23.10.1 minor release:

- Support Intel® Core™ Ultra processor with NDA packages
- Update Base Container components supported by On Prem Edge AI Box Configuration Profile

New Hardware (Platforms/CPUs/GPUs/Accelerators):

- Intel® Core™ Ultra processor for Edge Deployments
- 5th Gen Intel® Xeon® Scalable processor for the Edge/IoT

Removed Support:

- Officially discontinued support for 3rd Gen Intel® Xeon® Scalable processors

Known Limitations/Restrictions:

- CPU Control Plane Plugin for Kubernetes* is only supported on Ubuntu OS with single node deployment

[4] See backup for workloads and configurations or visit www.Intel.com/PerformanceIndex. Results may vary.

- MinIO* is supported only with CRI-O runtime
- Only in-tree Intel® QuickAssist Technology (Intel® QAT) drivers supported on RHEL 9.2 and Rocky 9.2
- UserSpace CNI with VPP is not supported
- Intel® Trust Domain Extensions (Intel® TDX) on VMRA does not support Intel® Dynamic Load Balancer (Intel® DLB), Intel® DSA, Intel® QAT, or network adapter device passthrough due to Intel® TDX driver security concerns
- Intel® Ethernet Operator DDP update feature might not work in rare cases; Legacy DDP update feature does not work
- KubeVirt and GPU_DP do not support Docker runtime
- Intel® Media Transport Library v23.08 only supports ICE versions between 1.9.11 and 1.11.14
- OneAPI ITEX sample does not support the GPU path on RHEL 9.2

Note: See GitHub for full details about Known Limitations.

A.2 Reference System 23.10 Release Notes

New Components/Features:

- Extended support for 5th Gen Intel® Xeon® Scalable processors to VMRA, which requires the NDA version of Intel® QuickAssist Technology (Intel® QAT) drivers
- Support for Intel® Trust Domain Extensions (Intel® TDX) on 5th Gen Intel® Xeon® Scalable processors with limitations
- Support for 12th Gen Intel® Core™ processor for IoT Edge
- Support for Edge AI Box 3.1 on 12th Gen Intel® Core™ desktop processors, 12th Gen Intel® Core™ processor for IoT Edge
- Extended support for Software Defined Factory use-case on Intel® Core™ and Intel Atom® platforms to include an additional platform to manage the deployment of controller node via Kubernetes
- Support deployment of a Machine Controller use case via VMRA
- Support ECDSA keys for 5G O-RAN security with NETCONF server/client authentication (Intel® Software Guard Extensions (Intel® SGX))
- Added boot guard and secure boot to 5G O-RAN security
- Use of Vector Packet Processing (VPP) (23.06) dataplane within Calico
- Integrated Intent-Driven Orchestration Release v0.2 with BMRA
- Support for Intel® Infrastructure Processing Unit (Intel® IPU) ASIC E2000 card and Infrastructure Programmer Development Kit (IPDK) Networking Recipe
- Support for virtual machine and bare metal mixed Kubernetes deployment
- Support for generic virtual machine type
- Integrated Intel® Media Transport Library library
- Integrated Intel® XPU Manager (Intel® XPUM) for Intel data center GPU health and telemetry monitoring
- Added Rocky Linux 9.2 as base OS

Updates/Changes:

- Version upgraded for the vast majority of RA components (See Section 4.1 for all supported versions)
- Notable updates:
  - FlexRAN™ to v23.07
  - Kubernetes* to v1.27.1
  - Intel® Containerized Telegraf to 1.3
  - Service Mesh Istio to v1.19.0
  - Intel® Managed Distribution of Istio* Service Mesh to v1.19.0-intel.0
  - Open vSwitch with DPDK to 3.2
  - Intel Multi-Buffer Crypto for IPsec library to v1.4
  - Intel® QAT Engine for OpenSSL to v1.3.1
  - Intel® Power Manager to v2.3.1
  - Data Plane Development Kit (DPDK) to v23.07
  - Intel® Ethernet Operator v23.08
  - SR-IOV FEC Operator to v2.7.1
  - OpenVINO™ v2023.1
  - Key Management Reference Application (KMRA) v2.4
  - Kubernetes to v1.27 for Cloud RA on Microsoft* Azure Kubernetes Service (AKS) and Amazon Elastic Kubernetes Service* (Amazon EKS)

Updates/Changes made for the Reference System 23.07.1 minor release:

- FlexRAN™ Docker container support (timer and XRAN mode test) on 5th Gen Intel® Xeon® Scalable processors (MCC)
- Support for Edge AI Box through new configuration profile - on_prem_aibox
- Integrated Docker Compose 2.12
- Generated Base Containers Dockerfiles for Edge AI Box, including

    aibox-base
      GPU user mode drivers and runtimes (202230714)
      OpenVINO™ runtime (2022.3)
    aibox-base-dev
      OpenVINO™ developer tools (2022.3)
    aibox-dlstreamer
      DLStreamer (2022.3)
    aibox-opencv-ffmpeg
      OpenCV (4.7)
      FFMPEG (cartwheel 2023q1)

  - Bug fix in Project Sylva validation suite to revise device ID map and readme and config update

New Hardware (Platforms/CPUs/GPUs/Accelerators):

- 12th Gen Intel® Core™ desktop processors with Intel® Arc™ Discrete Graphics GPU A380
- 12th Gen Intel® Core™ processor for IoT Edge with Intel® Iris® X^e Integrated Graphics
- 13th Gen Intel® Core™ mobile processor with Intel® Iris® X^e Integrated Graphics

Removed Support:

- Discontinued support for Rocky 9.1 as base OS

Known Limitations/Restrictions:

- Intel® Data Center GPU Flex Series, CPU Control Plane Plugin for Kubernetes*, Intel® Media SDK(only Docker runtime) are only supported on Ubuntu OS
- MinIO* is supported only with CRI-O runtime
- Only in-tree Intel® QuickAssist Technology (Intel® QAT) drivers supported on RHEL 9.2 and Rocky 9.2
- Intel® Data Streaming Accelerator (Intel® DSA) may not work on some older (earlier stepping) CPUs on RHEL 9.2 and Rocky Linux 9.2
- Work-around applied to fix default Intel® QAT driver 4.23.0 install fail in VMRA mode on 3rd Gen Intel® Xeon® Scalable processors
- UserSpace CNI with VPP is not supported
- Intel® Trust Domain Extensions (Intel® TDX) on VMRA does not support Intel® Dynamic Load Balancer (Intel® DLB), Intel® DSA, Intel QAT, or network adapter device passthrough due to Intel® TDX driver security concerns
- Intel Ethernet Operator DDP update feature might not work in rare cases; use legacy DDP update feature instead
- Mixed deployment `on_prem_sw_defined_factory` profile with 12th Gen Intel® Core™ or later CPU should disable hyper thread

Note: See GitHub for full details about Known Limitations.

A.3 Reference System 23.07 Release Notes

New Components/Features:

- Support for 5th Gen Intel® Xeon® Scalable processors requires the NDA version of the Intel® QuickAssist Technology (Intel® QAT) drivers
- Support for 12th Gen Intel® Core™ processors
- Support for Intel Atom® x6000e series processors
- Support for 5G O-RAN security with NETCONF server/client authentication (Intel® Software Guard Extensions (Intel® SGX))
- Support Rancher RKE2 Kubernetes distribution for the access edge configuration profile with Ubuntu (v1.26.2+rke2r1)
- Support for Intel® FPGA SmartNIC WSN6050 Platform for video production
- Enabled Intel® Infrastructure Processing Unit (Intel® IPU) ASIC E2000 card and made it available in the host machine, which requires the NDA version of image
- Support for Intel® SGX in VMRA by upgrading QEMU and libvirt
- Support for Key Management Reference Application (KMRA) in Virtual Machine Reference Architecture (VMRA)
- Support for KMRA on 4th Gen Intel® Xeon® Scalable processors on production SKUs
- Implemented Intel® SGX-enabled Istio in VMs
- Support for Cilium eBPF Dataplane on Microsoft Azure Kubernetes Service (AKS)
- Updates to Kubernetes version and tools used to deploy on Microsoft Azure Kubernetes Service (AKS) and Amazon Web Service (AWS) Elastic Kubernetes Service (EKS) in Cloud Reference System Architecture
- Implemented support and option for Intel® QuickAssist Technology (Intel® QAT) in-tree versus out-of-tree drivers and libraries
- Integrated Intel® oneAPI Base Toolkit (Base Kit) 2023.1 and Intel® AI Analytics Toolkit (AI Kit) 2023.1.1
- Integrated FFmpeg with cartwheel (Intel GitHub 2023q1 release)
- Added two Configuration Profiles:

  - On-Premises SW Defined Factory Configuration Profile for industrial use cases
  - On-Premises VSS Configuration Profile for Video Structuring Server (VSS) use cases
- RHEL 9.2 as base OS
- RHEL 9.2 RT as base OS
- Ubuntu 22.04.2 as base OS

Updates/Changes:

- Version upgraded for the vast majority of Reference System components (See elsewhere in this document for complete BOM and versions)
- Notable updates:
  - FlexRAN™ to v23.03
  - Kubernetes* to v1.26.3
  - CPU Control Plane Plugin for Kubernetes* to v0.1.2
  - Telemetry Aware Scheduling to v0.5.0
  - GPU Aware Scheduling to v0.5.2
  - Intel® Power Manager to v2.2
  - Service Mesh Istio to v1.18.1
  - Intel® Managed Distribution of Istio* Service Mesh to v1.18.0-intel.0
  - Data Plane Development Kit (DPDK) to v23.05
  - Open vSwitch with DPDK to 3.11
  - Traffic Analytics Development Kit (TADK) to 23.03
  - OpenSSL to openssl-3.1.0
  - Intel® Data Center GPU Flex Series driver to 20230519 release
  - SR-IOV FEC Operator to 2.7
  - Intel® Platform Telemetry Insights to 23.05 (with license)
- Kubespray* is provided via ansible-galaxy collection instead of git submodule

Updates/Changes made for the Reference System 23.02.1 minor release:

- Stack Validation:
  - Test cases created for Anuket Reference Architecture Kubernetes Component Level Architecture specifications, to be used for Anuket Reference Conformance Kubernetes and Project Sylva Stack validation
  - Test cases created for Device Plugins Single Root IO Virtualization (SR-IOV) Data Plane Development Kit and Multus Container Network Interface
  - Cloud Native Network Function (CNF) Validation:
  - Test case to check CNF allocation of SR-IOV devices like virtual functions of network adapters or accelerators, to be used for Project Sylva CNF validation
- Added workaround for building the Intel® Ethernet Operator and SR-IOV FEC (Forward-Error Correction) Operator
- Resolved issue regarding the user-space CNI by disabling Vector Packet Processing (VPP)
- Removed dependency of Intel® QuickAssist Technology (Intel® QAT) on OpenSSL to allow independent deployment of Crypto libraries

New Hardware (Platforms/CPUs/GPUs/Accelerators):

- 5th Gen Intel® Xeon® Scalable processors (XCC, MCC)
- 4th Gen Intel® Xeon® Scalable processor with Intel® vRAN Boost up to 32 cores
- Intel® FPGA SmartNIC WSN6050 Platform
- 12th Gen Intel® Core™ processors
- Intel Atom® x6000e series processors
- Intel® Infrastructure Processing Unit (Intel® IPU) ASIC E2000 card

Removed Support:

- Discontinued supporting Cloud Native Data Plane (CNDP)
- Discontinued supporting RHEL 9.0 as base OS
- Discontinued supporting RHEL 8.6 RT as base OS

Known Limitations/Restrictions:

- Intel® Data Center GPU Flex Series, CPU Control Plane Plugin for Kubernetes, Intel® Media SDK (only Docker runtime) are only supported on Ubuntu OS
- FlexRAN™ container support is limited to v22.07, Ubuntu 22.04 base OS, and only on 3rd Gen Intel® Xeon® Scalable processors
- MinIO is supported only with CRI-O runtime

- Only in-tree Intel® QuickAssist Technology (Intel® QAT) and Intel® Ethernet Network Adapter E810 drivers supported on RHEL 9.2
- Intel® Ethernet Network Adapter E810 in-tree driver does not support VF function on RHEL 9.2, which impacts XRAN mode test in FlexRAN™ application
- Intel® QuickAssist Technology (Intel® QAT) is not supported on Rocky Linux 9.1 on 5th Gen Intel® Xeon® Scalable processors
- Intel® Data Streaming Accelerator (Intel® DSA) may not work on some older (earlier stepping) CPUs on RHEL 9.2
- UserSpace CNI with VPP is not supported
- Rancher only supported for containerd
- CAdvisor not supported on CRI-O runtime

Note: See GitHub for full details about Known Limitations.

A.4 Reference System 23.02 Release Notes

New Components/Features:

- Media Analytics Libraries
  - Intel® Deep Learning Streamer (Intel® DL Streamer), GStreamer, OpenVINO™ toolkit
  - OpenCL™ software, Level zero GPU, DPC++, and VAAPI from the Intel® GPU toolkit
- FlexRAN™ software running as a Docker container (now available without NDA)
- Rook/Ceph as a storage-related component
- Rocky Linux 9.1 as base operating system (with some limitations mentioned below)
- Non-root user deployment of Virtual Machine Reference System Architecture (VMRA)
- Custom cluster naming in VMRA
- Support for using Amazon Web Services (AWS) and Azure "Cloud" CLIs as an alternative to Terraform
- Azure Kubernetes Service (AKS) support for static CPU Management Policy and Intel® CPU Control Plane Plugin for Kubernetes
- Intel® Software Guard Extensions (Intel® SGX) on AKS

Updates/Changes:

- Software versions upgraded for the majority of Reference System components (See User Guide for complete BOM and versions)

  Notable updates:
  - Kubernetes to v1.26.1
  - MinIO to v4.5.8
  - DPDK to v22.11.1
  - Service Mesh to v1.17.1
  - VPP to v2302
  - KMRA to v2.3
- Eliminated the BMRA for Object Storage Setup deployment model. The storage-related features (MinIO, LPVSP, and Rook/Ceph) are now provided as optional components in select Configuration Profiles.
- Support of geo-specific mirrors for Kubespray (for example, in the People's Republic of China)
- Supported Kubernetes versions updated for AKS and Amazon EKS
- Ubuntu images updated for AKS and Amazon EKS
- Ability to deploy more Reference System software components on Azure and AWS
  - Elasticsearch
  - Kibana

Updates/Changes made for the Reference System 23.11.1 minor release:

- Intel® QAT 2.0 drivers for 4th Gen Intel® Xeon® Scalable processors (formerly code named Sapphire Rapids [SPR]) are sourced from public repo and no longer under NDA. Ignore Guide requirement to provide the *QAT20.L.0.9.9-00019.tar.gz* driver package file.
- Resolved issue regarding downloading CPUID for Rocky Linux 8.5 and RHEL 9.

New Hardware (Platforms/CPUs/GPUs/Accelerators):

- N/A

Removed Support:

- `full_nfv` profile
- Ubuntu 20.04 as base operating system
- Rocky Linux 9.0 as base operating system

Known Limitations/Restrictions:

- When using the Cilium CNI, secondary interfaces are not supported

- Intel® Dynamic Load Balancer (Intel® DLB) is not fully supported on Rocky Linux 9.1
- FlexRAN container support is limited to FlexRAN v22.07, Ubuntu 22.04 base operating system, and only on 3rd Gen Intel® Xeon® Scalable processors
- Media Analytics is supported only with Docker runtime
- MinIO is supported only with CRI-O runtime
- VMRA cluster expansion with additional VM nodes might fail
- Trusted Certificate Attestation (TCA) is not fully functional in VMRA

The following table lists key features of the 4th Gen Intel Xeon Scalable processor and the support for those features in Reference System 23.02.

Table 23. Status of Support for Key Features of 4th Gen Intel Xeon Scalable Processor in BMRA 23.02

Category	Feature	BMRA 23.02 Support	BMRA 23.02 Status/Comments
CPU / Accelerator	IAX	Yes	
	QAT	Yes	
	DLB	Yes	Not yet available through hypervisor
	DSA	Yes	Not yet available through hypervisor
Power Management	SST-PP, SST-TF SST-BF, SST-CP	Yes	
Security	SGX	Yes	
RAS	RAS	Yes	
ISA	FP-16 (5G ISA)	Yes	
	AMX (TMUL)	No	Not yet supported in Reference System
	VP2INTERSECT	Yes	
	AIA (MOVDIRI, Power Instrs.)	Yes	
I/O	CXL 1.1	Yes	
	PCI Gen5	Yes	
Virtualization	Intel® Scalable IOV	Yes	
	SVM	Yes	Supported for 4th Gen Intel® Xeon® Scalable processor

Refer to the following tables for other features of 4th Gen Intel Xeon Scalable processor enabled in prior BMRA releases.

Appendix B Abbreviations

The following abbreviations are used in this document.

Abbreviation	Description
AGF	Access Gateway Function
AI	Artificial Intelligence
AIC	Add In Card
AIA	Accelerator Interfacing Architecture
AMX	Advance Matrix Multiply
API	Application Programming Interface
BIOS	Basic Input/Output System
BKC	Best Known Configuration
BMRA	Bare Metal Reference Architecture
BOM	Bill of Material
CA	Certificate Authority
CDN	Content Delivery Network
CLOS	Class of Service
Cloud RA	Cloud Reference System Architecture

Abbreviation	Description
CMK	CPU Manager for Kubernetes
CMTS	Cable Modem Termination System
CNCF	Cloud Native Computing Foundation
CNDP	Cloud Native Data Plane (CNDP)
CNI	Container Network Interface
CO	Central Office
CTK	Crypto-API Toolkit
CU	Central Unit
CXL	Compute Express Link
DDP	Dynamic Device Personalization
DHCP	Dynamic Host Configuration Protocol
DLB	Intel® Dynamic Load Balancer (Intel® DLB)
DNS	Domain Name Service
DPDK	Data Plane Development Kit
DRAM	Dynamic Random Access Memory
DSA	Intel® Data Streaming Accelerator (Intel® DSA)
DU	Distribution Unit
EIST	Enhanced Intel SpeedStep® Technology
FPGA	Field-Programmable Gate Array
FW	Firmware
GAS	GPU Aware Scheduling
GPU	Graphics Processor Unit
HA	High Availability
HCC	High Core Count
HSM	Hardware Security Model
HT	Hyper Threading
IAX	In-Memory Analytics
IMC	Integrated Memory Controller
Intel® AVX	Intel® Advanced Vector Extensions (Intel® AVX)
Intel® AVX-512	Intel® Advanced Vector Extension 512 (Intel® AVX-512)
Intel® DCAP	Intel® Software Guard Extensions Data Center Attestation Primitives (Intel® SGX DCAP)
Intel® DLB	Intel® Dynamic Load Balancer (Intel® DLB)
Intel® DSA	Intel® Data Streaming Accelerator (Intel® DSA)
Intel® HT Technology	Intel® Hyper-Threading Technology (Intel® HT Technology)
Intel® QAT	Intel® QuickAssist Technology (Intel® QAT)
Intel® RDT	Intel® Resource Director Technology (Intel® RDT)
Intel® SecL – DC	Intel® Security Libraries for Data Center (Intel® SecL – DC)
Intel® SGX	Intel® Software Guard Extensions (Intel® SGX)
Intel® SST-BF	Intel® Speed Select Technology – Base Frequency (Intel® SST-BF)
Intel® SST-CP	Intel® Speed Select Technology – Core Power (Intel® SST-CP)
Intel® SST-PP	Intel® Speed Select Technology – Performance Profile (Intel® SST-PP)
Intel® SST-TF	Intel® Speed Select Technology – Turbo Frequency (Intel® SST-TF)
Intel® TDX	Intel® Trust Domain Extensions (Intel® TDX)
Intel® VT-d	Intel® Virtualization Technology (Intel® VT) for Directed I/O (Intel® VT-d)
Intel® VT-x	Intel® Virtualization Technology (Intel® VT) for IA-32, Intel® 64 and Intel® Architecture (Intel® VT-x)
IOMMU	Input/Output Memory Management Unit

Abbreviation	Description
IoT	Internet of Things
ISA	Instruction Set Architecture
I/O	Input/Output
K8s	Kubernetes
KMRA	Key Management Reference Application (KMRA)
KMS	Key Management Service (KMS)
LCC	Low Core Count
LLC	Last Level Cache
LOM	LAN on Motherboard
LPVSP	Local Persistent Volume Static Provisioner
MEC	Multi-Access Edge Compute
mTLS	Mutual Transport Layer Security
NFD	Node Feature Discovery
NFV	Network Function Virtualization
NIC	Network Interface Card (Network Adapter)
NTP	Network Time Protocol
NUMA	Non-Uniform Memory Access
NVM/NVMe	Non-Volatile Memory
OAM	Operation, Administration, and Management
OCI	Open Container Initiative
OS	Operating System
OVS	Open vSwitch
OVS DPDK	Open vSwitch with DPDK
PBF	Priority Based Frequency
PCCS	Provisioning Certification Caching Service
PCI	Physical Network Interface
PCIe	Peripheral Component Interconnect Express
PF	Port Forwarding
PMD	Poll Mode Driver
PMU	Power Management Unit
PXE	Preboot Execution Environment
QAT	Intel® QuickAssist Technology
QoS	Quality of Service
RAS	Reliability, Availability, and Serviceability
RDT	Intel® Resource Director Technology
RHEL	Red Hat Enterprise Linux
S3	Amazon Web Services Simple Storage Service
S-IOV	Intel® Scalable I/O Virtualization (Intel® Scalable IOV)
SA	Service Assurance
SGX	Intel® Software Guard Extensions (Intel® SGX)
SR-IOV	Single Root Input/Output Virtualization
SSD	Solid State Drive
SSH	Secure Shell Protocol
SVM	Shared Virtual Memory
TADK	Traffic Analytics Development Kit
TAS	Telemetry Aware Scheduling

Abbreviation	Description
TCA	Trusted Certificate Attestation
TCS	Intel® Trusted Certificate Service
TCO	Total Cost of Ownership
TDP	Thermal Design Power
TLS	Transport Layer Security
TME	Total Memory Encryption
TMUL	Tile Multiply
UEFI	Unified Extensible Firmware Interface
UPF	User Plane Function
vBNG	Virtual Broadband Network Gateway
vCDN	Virtualized Content Delivery Network
vCMTS	Virtual Cable Modem Termination System
VF	Virtual Function
VMRA	Virtual Machine Reference Architecture
VPP	Vector Packet Processing
vRAN	Virtual Radio Access Network
VSS	Video Structuring Server
WAF	Web Application Firewall

Performance varies by use, configuration and other factors. Learn more at www.Intel.com/PerformanceIndex.

Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. See backup for configuration details. No product or component can be absolutely secure.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

Intel technologies may require enabled hardware, software or service activation.

Intel does not control or audit third-party data. You should consult other sources to evaluate accuracy.

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Code names are used by Intel to identify products, technologies, or services that are in development and not publicly available. These are not "commercial" names and not intended to function as trademarks.

0124/DN/WIT/PDF 721796-012US