USER GUIDE

Intel Corporation

Network and Cloud Edge Virtual Machine Reference System Architecture Release v22.05

Authors

Abhijit Sinha

Calin Gherghe

Dana Nehama

Daniel Ugarte

Francis Cahill

Joel A. Gibson

Michael O'Reilly

Octavia Carotti

1 Introduction

1.1 Purpose and Scope

The **Virtual Machine Reference System Architecture (VMRA)** is part of the Network and Cloud Edge Reference System Architectures Portfolio. The VMRA is a common virtual cluster template platform. It is composed of a set of virtual machines, implemented on a single physical Intel node or multi-nodes, that can be used for hosting a Kubernetes cluster.

Network locations (for example, On-Premises Edge and Remote Central Office) require deployment of different hardware, software, and configuration specifications due to varying workloads, cost, density, and performance requirements. Configuration Profiles define prescribed sets of VMRA hardware and software components designed to optimally address the diverse deployment needs. Ansible playbooks implement the Configuration Profiles for fast, automatic deployment of needed VMRA capabilities. The result is an optimized installation of the VMRA Flavor as defined by the selected Configuration Profile. This user guide covers implementation of VMRA using several Configuration Profiles for network-location-specific and generic deployments.

Network-location-specific Configuration Profiles covered in this document include:

- **On-Premises Edge Configuration Profile** –Typical Customer Premises deployment supporting, for example, Content Delivery Network (CDN) and Smart City scenarios.
- **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. Currently tailored exclusively for Media Visual Processing workloads such as CDN Transcoding.

Other Configuration Profiles enable flexible deployments and include the following:

- **Basic Configuration Profile** – A generic minimum VMRA Kubernetes cluster setup.
- **Full Configuration Profile** – A generic complete VMRA setup based on all software features.
- **Build-Your-Own Configuration Profile** – A complete set of all available software features targeted at developers and deployers who are looking to evaluate, control, and configure all the software and hardware ingredients and dependencies individually.

More information on Configuration Profiles and implementation of VMRA Flavors using the Configuration Profiles is provided later in this document.

1.2 User Guide Information

This document contains step-by-step instructions on installation, configuration, and use of networking and device plug-in features for deploying the VMRA Release v22.05 by implementing the VMRA template platform with the above Configuration Profiles . Validated, open source Ansible playbooks automatically provision the virtual environment along with a Kubernetes cluster (if desired) for the selected Configuration Profiles. enabling user to quickly and easily create predictable deployments.

By following this document, it is possible to set up a virtual cluster based on Kubernetes with optimized configurations for cloud native deployments.

This document provides the following information:

- Part 1 (Sections 2 – 5): Requirements for hardware and software to prepare for the Ansible scripts.
- Part 2 (Sections 6 – 12): Step-by-step instructions on how to build each VMRA Flavor by implementing the configuration profiles. **If you wish to start building the VMRA right away, you may directly go to these sections and start automatically provisioning the VMRA Flavor of your choice.**
- Part 3 (Appendix A): VMRA Release Notes
- Part 4 (Appendix B): Abbreviations

See the Network and Cloud Edge Reference System Architectures Portfolio User Manual for an overview of the Reference System Architectures.

1.3 Version 22.05 Release Information

The hardware platforms supported by VMRA v22.05 are based on the 2nd Generation Intel® Xeon® Scalable processors, 3rd Generation Intel® Xeon® Scalable processors, and Intel® accelerators. This release also supports an early access version of 4th Generation Intel® Xeon® Scalable processors code named Sapphire Rapids (SPR).

The supported software components comprise open source cloud-native software delivered by Intel and its partners for overcoming barriers to networking adoption and private/public cloud deployments.

Due to the hardware abstraction in the VMRA virtual setup, some hardware-dependent software features available in a Container Bare Metal Reference System Architecture (BMRA) are not supported by the VMRA. For details, about the technologies supported refer to the Network and Cloud Edge Reference System Architectures Portfolio User Manual.

Following are the primary new and updated hardware ingredients and software features introduced in Release v22.05. For additional details, refer to the VMRA Release Notes in this document.

Hardware Supported:

- Included the multi-node deployment model (previous single node deployment)
- Archer City (2S-SPR XCC, MCC) Quanta S6Q SDP (2S-SPR XCC, MCC), DSG's Fox Creek Pass (2S-SPR XCC, MCC), Ruby Pass (1S-SPR XCC, MCC)
- Power Management Unit (PMU) telemetry for power C-state for 4th Generation Intel® Xeon® Scalable processors

Configuration Profiles:

- Added Build-Your-Own Configuration Profile

Service Mesh

- Microservices architectures with Istio service mesh (using istioctl for deployment) and Envoy.

Packet Processing

- Support for Cloud Native Data Plane (CNDP) for enabling a cloud-based framework for networking and accelerated packet processing, provisioning, orchestrating, and managing microservices to scale efficiently with Kubernetes deployment.
- Updated Data Plane Development Kit (DPDK) to version 22.03

Operating System

- Rocky Linux 8.5
- Ubuntu 20.04LTS, 22.04LTS

Experience Kits, the collaterals that explain in detail the technologies enabled in VMRA release 22.05, including benchmark information, are available in the following locations:

- Network Transformation Experience Kits
- Container Experience Kits

For NDA material, contact your regional Intel representative.

Table of Contents

Figures

Tables

Document Revision History

REVISION	DATE	DESCRIPTION
001	February 2022	Initial release.
002	March 2022	Updated a few URLs.
003	June 2022	Covers the 4th Generation Intel® Xeon® Scalable processor (formerly code named Sapphire Rapids).
004	June 2022	Changes include updates to the Known Issues section.

1.4 Key Terms

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

Table 1. Terms Used

TERM	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 Cloud 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 IT needs. The portfolio offers two 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.
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. In addition, generic Configuration Profiles are available to developers for flexible deployments.
Reference 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 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 2nd Generation, 3rd Generation, and 4th Generation Intel® Xeon® Scalable processors. The platforms integrate Intel® Ethernet Controller 700 Series and 800 Series, Intel® QuickAssist Technology (Intel® QAT), Intel® Server GPU (Graphic Processor Unit), Intel® Optane™ technology, and more. ***Note:*** This release of VMRA does not support the Intel® Xeon® D processor.

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)

TERM	DESCRIPTION
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

1.5 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) • Key Management Reference Application (KMRA) implementation
Storage	Creating a disaggregated, high-performance, scalable storage platform using MinIO 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 Trusted Certificate Service for Kubernetes* platform, providing more secure Key Management.

1.6 Reference Documentation

The *Network and Cloud Edge Reference System Architectures Portfolio User Manual* contains a complete list of reference documents. Additionally, a bare metal-based reference system architecture (BMRA) deployment allows creation of a Kubernetes cluster on multiple nodes. The *Network and Cloud Edge Container Bare Metal Reference System Architecture User Guide* provides information and installation instructions for a BMRA.

Collaterals, including technical guides and solution briefs that explain in detail the technologies enabled in VMRA release v22.05, are available in the following locations: https://networkbuilders.intel.com/intel-technologies/network-transformation-exp-kits and https://networkbuilders.intel.com/intel-technologies/container-experience-kits.

Part 1:
Reference Architecture Deployment:
Ansible Playbooks
Common Hardware Components
Software Ingredients
Recommended Configurations

2 Reference Architecture Deployment

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

2.1 VMRA Architecture

The VMRA is a virtual cluster implemented on a single physical Intel node (Figure 1). The VMRA supports options for a virtual Kubernetes cluster or VMs cluster. A VMRA allows flexible deployment options for creating networking solutions for production or testing.

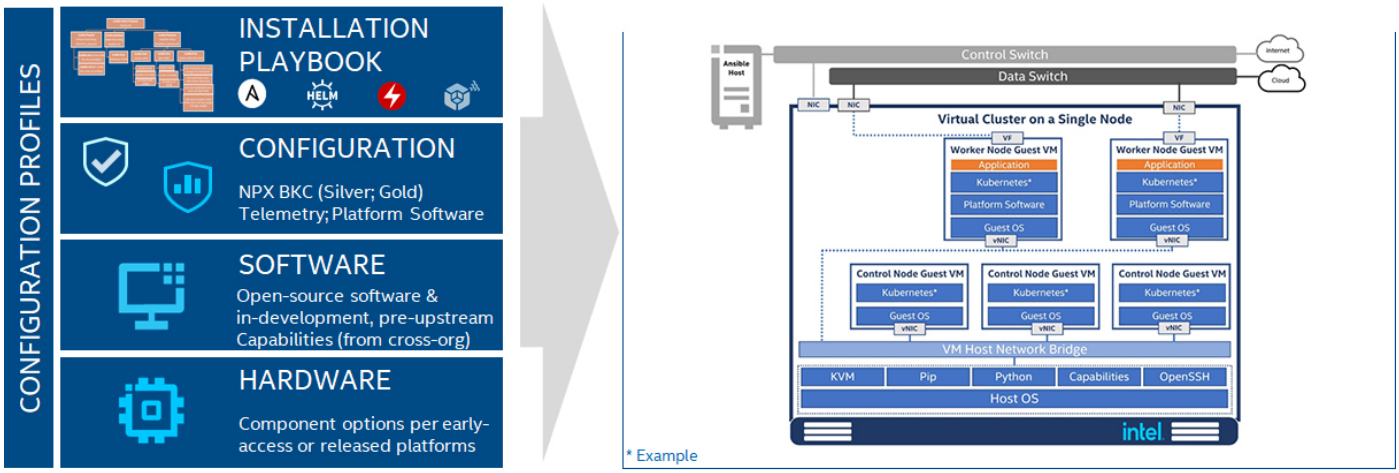

Figure 1 Virtual Machine Reference System Architecture Illustration with Kubernetes Cluster

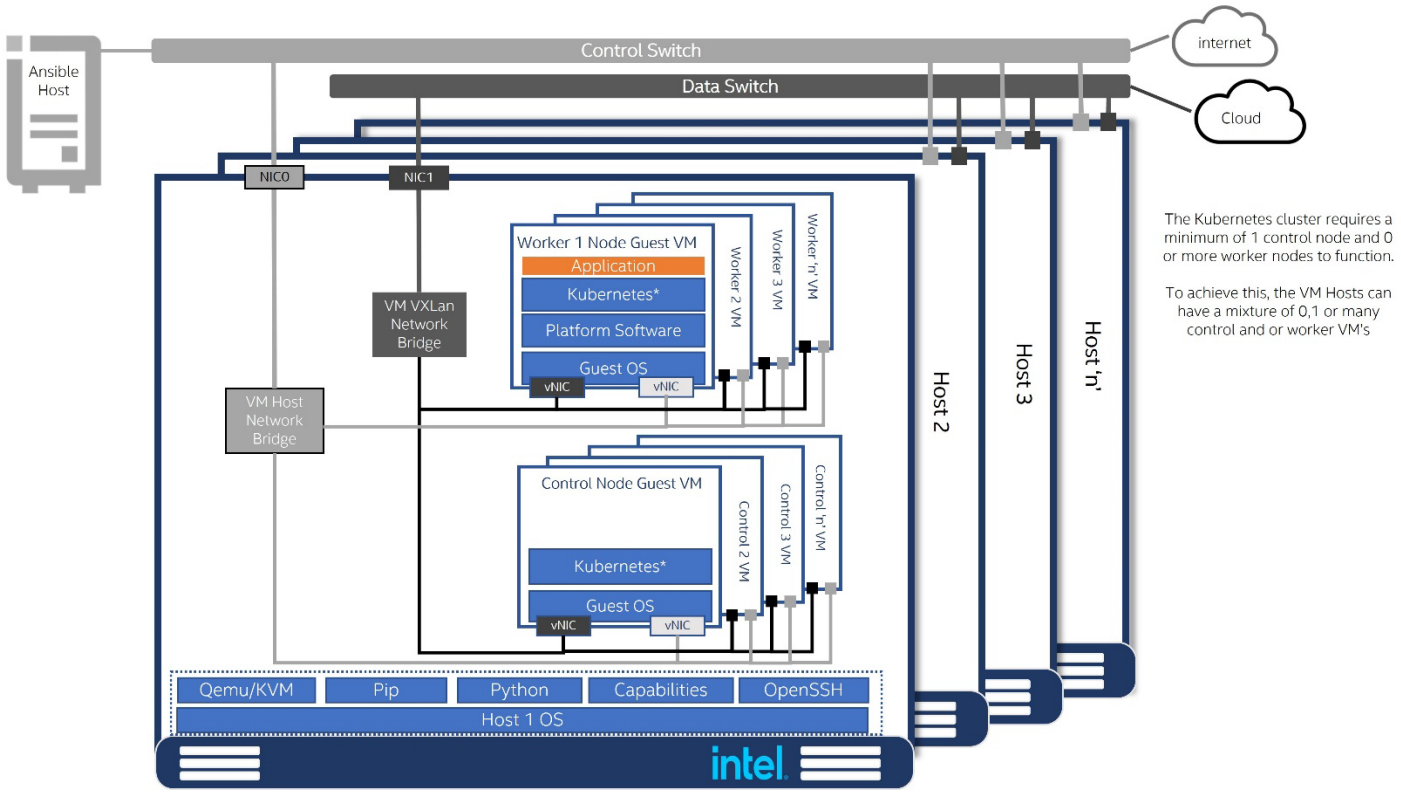

Figure 2. VMRA Multiple-Node Deployment

2.2 Configuration Profiles

A Configuration Profile describes specific hardware and software bill of materials (BOM) and configurations, applicable for a specific deployment. Configuration Profiles take into consideration the best-known configuration (BKC) validated by Intel for optimized performance.

installation scripts implement a VMRA Flavor by deploying the required components specified by a Configuration Profile. Each VMRA Flavor 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 VMRA deployment is populated with the maximum available network interface cards (NICs).
- **Software BOM** leverages the Intel platform and enables cloud-native adoption.
- **Installation (Ansible) Playbook** automates the installation of a Reference Architecture Flavor per a Configuration Profile specification.

Three of the Reference Architecture Configuration Profiles are network location-specific:

- **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, 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 SMTC deployments, CDN, and Ad-insertion.
- **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 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.

Three additional Reference Architecture Configuration Profiles that are not location-specific enable flexible deployments per need:

- **Basic Configuration Profile** – Minimum VMRA Kubernetes cluster setup.
- **Full Configuration Profile** – Complete VMRA setup supporting all available software features. This profile is targeting developers and deployers that are looking to evaluate, control, and configure the software and hardware ingredients and dependencies.
- **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 the software and hardware ingredients and dependencies individually.

2.3 Reference Architecture Installation Prerequisites

This section helps you get ready for running 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 and VM hosts. 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 VM host

After satisfying these prerequisites, Ansible playbooks for 2nd Generation Intel Xeon Scalable processors and 3rd Generation Intel Xeon Scalable processors can be downloaded directly from the dedicated GitHub* page (https://github.com/intel/container-experience-kits/releases) or cloned using the Git. Be sure to complete the software prerequisites below before downloading the Ansible playbooks. Request access to the Ansible playbooks for the 4th Generation Intel Xeon Scalable processor from your regional Intel representative.

2.3.1 Hardware BOM Selection and Setup for 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 configurations described in the following sections:

- VM host – Refer to the following sections for recommended host assembly:
  - Base host node – Review Section 3.1 to satisfy base performance characteristics.
  - Plus host node – Review Section 3.2 to satisfy plus performance characteristics.
- Configuration Profile BOM – See Sections 7 through 12 for details about hardware BOM selection and setup for your chosen Configuration Profile.

2.3.2 BIOS Selection for VM Host

Enter the UEFI or BIOS menu and update the configuration as listed in Section 6 and Table 16, which describe the BIOS selection in detail.

2.3.3 Operating System Selection for VM Host and VMs

The following Linux operating systems are supported for the VM host and VMs:

- Ubuntu 20.04 LTS (20.04.43)
- Ubuntu 21.10 (21.10)
- Ubuntu 22.04 (22.04)
- Rocky Linux 8.5

For the supported distribution, the base operating system install image is sufficient to be built using the "Minimal" option during installation. In addition, the following must be met:

- The VM host must have network connectivity to the Ansible host.
- SSH connections are supported. If needed, install SSH Server with the following commands (internet access is required):

```
# sudo apt update
# sudo apt install openssh-server
```

2.3.4 Network Interface Requirements for VM Host

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

- Internet network
  - Available for VMs through a Linux bridge on the host, providing internet connectivity through NAT
  - 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 for Kubernetes installs (This can be a shared interface with the internet network)
  - Available for VMs through a Linux bridge on the host, connected to other nodes through VXLAN
  - Kubernetes control and worker node inter-node communications (for Kubernetes installs)
  - Calico pod network runs over this network (for Kubernetes installs)
  - Configured to use a private static address
- Tenant data networks
  - Dedicated networks for traffic
  - SR-IOV enabled
  - Virtual function (VF) can be DPDK bound in pod

2.3.5 Software Prerequisites for Ansible Host and VM Host

Before deployment of the VMRA Ansible playbooks, the Ansible host must be prepared. To successfully run the deployment, perform the following tasks before you download the VMRA Ansible code.

Perform the following steps:

1. Log in to the Ansible host machine using SSH or your preferred method to access the shell on that machine.
2. Install packages on Ansible Host. The following example assumes that the host is running RHEL 8. Other operating systems may have slightly different installation steps:

```
$ yum install python3
$ yum install libselinux-python3
```

3. Enable passwordless login between all VM Hosts and Ansible Host.
   Create authentication SSH-Keygen keys on Ansible Host:

```
$ ssh-keygen
```

Upload generated public keys to all VM Hosts from Ansible Host:

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

2.4 Ansible Playbook

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

2.4.1 Ansible Playbooks Building Blocks

The following components make up the VMRA Ansible playbooks.

Note: Ansible playbooks for 2nd Generation Intel Xeon Scalable processors and 3rd Generation Intel Xeon Scalable processors are open source and available here.

Ansible playbooks for the 4th Generation Intel® Xeon® Scalable processor are available under NDA. To obtain these Ansible playbooks, contact your regional Intel representative.

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

- inventory.ini
- group_vars/all.yml
- host_vars/host-for-vms-1.yml
- host_vars/host-for-vms-2.yml # used in case of multi-node setup
- host_vars/vm-ctrl-1.yml
- host_vars/vm-work-1.yml
- host_vars/vm-work-2.yml # used in case of multi-node setup

For default values in these files, refer to the Configuration Profile-specific sections for VMRA installations: Section 7, VMRA Basic Configuration Profile Setup, Section 8, VMRA Full Configuration Profile Setup, Section 9, VMRA On-Premises Edge Configuration Profile Setup, Section 10, VMRA Remote Central Office-Forwarding Configuration Profile Setup, Section 11, VMRA Regional Data Center Configuration Profile Setup, and Section 12, VMRA Build-Your-Own Configuration Profile Setup.

Ansible Playbooks act as a user entry point and include all relevant Ansible roles and Helm charts. Top-level Ansible playbooks exist for each Configuration Profile, which allows lean use case-oriented cluster deployments. Each playbook includes only the Ansible roles and configuration files that are relevant for a given use case. See High Level Ansible Playbooks in Figure 3.

- playbooks/basic.yml
- playbooks/full_nfv.yml
- playbooks/on_prem.yml
- playbooks/remote_fp.yml
- playbooks/regional_dc.yml
- playbooks/build_your_own.yml

VMRA is deployed through a single playbook that utilizes one of the playbooks for the Configuration Profiles you will deploy. In addition, the VMRA playbook ensures that both the host and VMs are configured as part of the infrastructure setup.

- playbooks/vm.yml

Each of these playbooks encompasses **Ansible Roles** grouped into three main execution phases, which are depicted in Figure 3 and further explained in the next section:

- Infrastructure Setup
- Kubernetes Deployment
- Capabilities Setup

Note that several Capabilities Setup roles include nested Helm charts for easier deployment and lifecycle management of deployed applications, as well as a group of **Common Utility Roles** that provide reusable functionality across the playbooks.

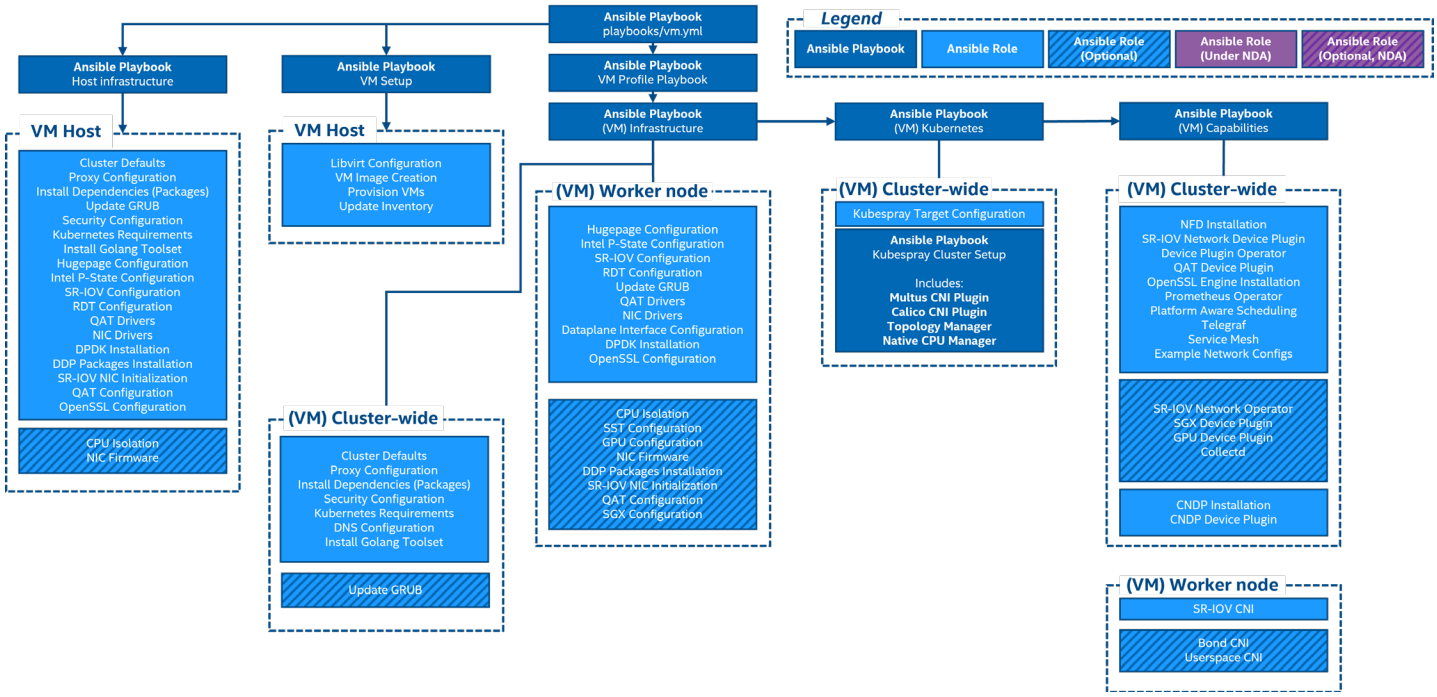

Figure 3. High Level VMRA Ansible Playbooks Architecture[1] Full Configuration Profile Example

2.4.2 Ansible Playbook Phases

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

1. **Host Infrastructure Setup** (sub-playbooks located in playbooks/infra/ directory)

   These playbooks modify kernel boot parameters and apply the initial system configuration for the host. Depending on the selected Configuration Profile, Host Infrastructure Setup includes:

   - Generic host OS preparation, for example, installation of required packages, Linux kernel configuration, proxy configuration, and modification of SELinux policies and firewall rules
   - Configuration of the kernel boot parameters according to the user-provided configuration 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 latest capabilities such as Dynamic Device Personalization (DDP) profiles are enabled
   - Installation of Dynamic Device Personalization 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. **Host Virtualization Setup** (playbooks/infra/prepare_vms.yml)

   This playbook installs and configures the virtualization layer and VMs that will be used as Kubernetes nodes later in the installation. Host Virtualization Setup includes:

   - Installing VM hypervisor and tools to manage VMs and images, such as QEMU, KVM, Libvirt, and Genisoimage
   - Create backing and configuration images for each VM
   - Create VXLAN bridges to ensure VMs connectivity cross multiple nodes
   - Start the VMs and perform optimization tasks (CPU pinning and NUMA alignment)
   - Collect information from VMs, make sure they are accessible
   - Update the Ansible Inventory to include VMs as controller and worker nodes according to the configuration

3. **VM Infrastructure Setup** (sub-playbooks located 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, VM Infrastructure Setup includes:

[1] Refer to https://www.intel.com/content/www/us/en/developer/get-help/legal-disclaimers-and-optimization-notices.html for more information regarding performance and optimization choices in Intel software products.

- Generic host OS preparation, for example, 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 to configure CPU isolation, hugepages, or explicitly enable/disable Intel P-state technology
- Configuration of SR-IOV and QAT devices
- Network Adapter drivers and firmware updates.

4. **Kubernetes Setup** (located in playbooks/k8s/ directory)

   This playbook deploys a high availability (HA) Kubernetes cluster using Kubespray. Kubespray is a project under the Kubernetes community that deploys production-ready Kubernetes clusters. The Multus CNI plugin, which is specifically designed to provide support for 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's also a container registry instance deployed to store images of various control-plane Kubernetes applications.

5. **VMRA System Capabilities Setup** (sub-playbooks located in 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 VMRA. 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.
   - CNI Plugins, which allow Kubernetes pods to be attached directly to accelerated and highly available hardware and software network interfaces.
   - 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.
   - Platform Aware Scheduling, which allows scheduling of workloads based on telemetry data.
   - Full Telemetry Stack consisting of Telegraf, Kube-Prometheus, and Grafana, which gives cluster and workload monitoring capabilities and acts as a source of metrics that can be used in TAS to orchestrate scheduling decisions.

2.5 Deployment Using Ansible Playbook

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

2.5.1 Prepare VM Host Server

For the VM host server, you must make sure that it meets the following requirements:

- Python 3 is installed. The version depends on the target distribution.
- SSH keys are exchanged between the Ansible host and VM host.
- Internet access on the VM host is mandatory. Proxies are supported and can be configured in the Ansible vars.
- BIOS configuration matching the desired state is applied. For details, refer to the specific Configuration Profile section for your profile: Section 7, VMRA Basic Configuration Profile Setup, Section 8, VMRA Full Configuration Profile Setup, Section 9, VMRA On-Premises Edge Configuration Profile Setup, Section 10, VMRA Remote Central Office-Forwarding Configuration Profile Setup, Section 11, VMRA Regional Data Center Configuration Profile Setup, and Section 12, VMRA Build-Your-Own Configuration Profile Setup.

For detailed steps on how to build the Ansible host, refer to Section 6.1.

2.5.2 Get Ansible Playbook and Prepare Configuration Templates

Perform the following steps:

1. Log in to your Ansible host (the one that you will run these Ansible playbooks from).
2. Clone the source code and change working directory.

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

3. Check out the latest version of the playbooks – using the tag from Table 18, for example:

```
git checkout v22.05
```

   Note: Alternatively go to https://github.com/intel/container-experience-kits/releases, download the latest release tarball, and unarchive it:

```
wget https://github.com/intel/container-experience-kits/archive/v22.05.tar.gz
tar xf v22.05.tar.gz
cd container-experience-kits-22.05
```

4. Initialize Git submodules to download Kubespray code.

```
git submodule update --init
```

5. Decide which Configuration Profile you want to deploy and export the environmental variable.
   For Kubernetes **Basic** Configuration Profile deployment:

   ```
   export PROFILE=basic
   ```

   For Kubernetes **Full** Configuration Profile deployment:

   ```
   export PROFILE=full_nfv
   ```

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

   ```
   export PROFILE=on_prem
   ```

   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 **Buil-Your-Own** Configuration Profile r deployment:

   ```
   export PROFILE=build_your_own
   ```

6. Install additional requirements.

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

7. Generate example profiles.

   ```
   make vm-profile ARCH=<skl,clx,icx,spr> NIC=<fvl,cvl> PROFILE=$PROFILE
   ```

2.5.3 Update Ansible Inventory File

Perform the following steps:

1. Edit the inventory.ini file generated in the previous steps.
   a. In the section [all], specify the target VM host server with hostname and Management IP address. Also set ansible_user to root (non-root user is not currently supported) and ansible_password to match the SSH configuration of the VM host. If the server is configured with passwordless SSH the ansible_password host variable can be removed.

   Note: The hostname can be the actual or a logical hostname. If a different hostname is used, be sure to update the configuration files such that host_vars/<hostname>.yml exists.

   b. In the [vm_host] section, update the hostname to match that defined in [all].

```
[all]
host-for-vms-1 ansible_host=10.0.0.1 ip=10.0.0.1 ansible_user=root ansible_password=XXXX
localhost ansible_connection=local ansible_python_interpreter=/usr/bin/python3

[vm_host]
host-for-vms-1

[kube_control_plane]
#vm-ctrl-1

[etcd]
#vm-ctrl-1

[kube_node]
#vm-work-1

[k8s_cluster:children]
kube_control_plane
kube_node

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

Do not uncomment any of the hostnames defined under [kube_control_plane], [etcd] and [kube_node], as these will be dynamically updated based on the number of virtual machines defined for the target VM host server in host_vars.

2.5.4 Update Ansible Host and Group Variables

Perform the following steps.

1. Create host_vars/<hostname>.yml for the target VM host server, matching the hostname from the inventory file. The provided host_vars/host-for-vms-1.yml can be copied to simplify this process:
   ```
   cp host_vars/host-for-vms-1.yml host_vars/<hostname>.yml
   ```
   In case of multi-node setup use host_vars/host-for-vms-2.yml as a template for all other VM hosts except the first one.
2. Update "vms" in host_vars/<hostname>.yml to match the desired number of VMs and their configuration. Note the "name" and "type" assigned to each VM, as these will be used to define host variables for each VM.

Note: For SR-IOV or QAT functionality, the VF PCI devices must be defined for each VM. This requires that the BDF (Bus:Device.Function) IDs are known prior to deploying the cluster. For more details, see the VM case configuration guide.

3. Create host_vars/<VM_name>.yml files for all VMs of type "work" defined in the previous step. The provided host_vars/vm-work-1.yml file can be copied to simplify this process:
   ```
   cp host_vars/vm-work-1.yml host_vars/<VM_name>.yml
   ```
4. Edit host_vars/<hostname>.yml, host_vars/<VM_name>.yml and group_vars/all.yml files to match your desired configuration. Each Configuration Profile uses its own set of variables.

Refer to the specific Configuration Profile section to get a full list of variables and their documentation: Section 7, VMRA Basic Configuration Profile Setup, Section 8, VMRA Full Configuration Profile Setup, Section 9, VMRA On-Premises Edge Configuration Profile Setup, Section 10, VMRA Remote Central Office-Forwarding Configuration Profile Setup, Section 11, VMRA Regional Data Center Configuration Profile Setup, and Section 12, VMRA Build-Your-Own Configuration Profile Setup.

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.

It is recommended that you check dependencies of components enabled in group_vars and host_vars with the packaged dependency checker:

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

Note: This will only run the dependency checker against the VM host. The check will be run against the VM configurations during deployment

Otherwise, you can skip directly to your chosen Configuration Profile playbook:

```
ansible-playbook -i inventory.ini playbooks/vm.yml
```

Note: The configuration profile is based on the PROFILE environment variable, which should match the profile that was used while generating the templates.

After the playbook finishes without any "Failed" tasks, you can proceed with the deployment validation described in Section 5 Post Deployment Verification Guidelines.

Note: Additional information can be found in the Ansible Playbook readme.

3 Reference Architecture Hardware Components and BIOS

For all Configuration Profiles, this section provides a menu of all possible hardware components for the VM Host as well as the BIOS components available.

3.1 Hardware Component List for Host Base

The following tables list the hardware options for the host in the "base" configuration. If your configuration needs improved processing, you may choose to use the "plus" configuration instead. See the next section for details.

Table 4. Hardware Components for Host Base – 2nd Generation Intel Xeon Scalable Processor

INGREDIENT	REQUIREMENT	REQUIRED/ RECOMMENDED
2nd Generation Intel® Xeon® Scalable processors	Intel® Xeon® Gold 6230 processor @ 2.1 GHz or 6230N CPU @ 2.3 GHz 20 C/40 T, 125 W, or higher number Intel® Xeon® Gold or Platinum CPU SKU	Required
Memory	Option 1: DRAM only configuration: 384 GB (12 x 32 GB DDR4 2666 MHz)	Required
	Option 2: DRAM only configuration: 384 GB (24 x 16 GB DDR4 2666 MHz)	
	Option 3: DRAM + Intel® Optane™ Persistent Memory DRAM: 192 GB (12x 16 GB DDR4, 2666 MHz)	
Intel® Optane™ Persistent Memory	512 GB (4x 128 GB Intel® Optane™ persistent memory in 2-1-1 Topology)	Recommended
Network Adapter	Option 1: Dual Port 100 GbE Intel® Ethernet Network Adapter E810-CQDA2 QSFP28	Required
	Option 2: Intel® Ethernet Network Adapter XXV710-DA2 QSFP28	
Intel® QAT	Intel® QuickAssist Adapter 8970 (PCIe) AIC or equivalent third-party Intel® C620 Series Chipset Intel® QAT enabled with minimum 8 lanes of PCIe connectivity	Required
Storage (Boot Drive)	Intel® SATA Solid State Drive D3 S4510 at 480 GB or equivalent boot drive	Required
Storage (Capacity)	Intel® NVMe SSD DC P4510 Series P4510 at 2 TB or equivalent (Recommended NUMA Aligned)	Required
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	Required
Additional Plug-in cards	N/A	

Table 5. Hardware Components for Host Base – 3rd Generation Intel Xeon Scalable Processor

INGREDIENT	REQUIREMENT	REQUIRED/ RECOMMENDED
3rd Generation Intel Xeon Scalable processors	Intel® Xeon® Gold 5318N processor at 2.1 GHz, 24 C/48 T, 150 W, or higher number Intel® Xeon® Gold or Platinum CPU SKU	Required
Memory	Option 1: DRAM only configuration: 256 GB (8 x 32 GB DDR4, 2666 MHz)	Required
	Option 2: DRAM only configuration: 256 GB (16 x 16 GB DDR4, 2666 MHz)	
Intel® Optane™ Persistent Memory	512 GB (4x 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-XXVDA-2	
Intel® QAT	Intel® QuickAssist Adapter 8960 or 8970 (PCIe*) AIC or equivalent third-party Intel® C620 Series Chipset	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
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	Required
Additional Plug-in cards	N/A	

Table 6. Hardware Components for Host Base – 4th Generation Intel Xeon Scalable Processor

INGREDIENT	REQUIREMENT	REQUIRED/ RECOMMENDED
4th Generation Intel Xeon Scalable processors	Intel® Xeon® Gold 5418N processor at 2.0GHz, 24 C/ 48 T, 165 W	Required
Memory	DRAM only configuration: 256 GB DRAM (16x 16 GB DDR5)	Required
Intel® Optane™ Persistent Memory	512 GB (4x 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-XXVDA-2	
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)	Required
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	Required
Additional Plug-in cards	N/A	

3.2 Hardware Component List for Host Plus

The following tables list the hardware options for the Host in the "plus" configuration, which helps improves the processing capability due to more powerful CPU, more memory, more disk space, and an amazingly fast network.

Table 7. Hardware Components for Host Plus – 2nd Generation Intel Xeon Scalable Processor

INGREDIENT	REQUIREMENT	REQUIRED/ RECOMMENDED
2nd Generation Intel® Xeon® Scalable processors	Intel® Xeon® Gold 6252 processor @ 2.1 GHz or 6252N processor @ 2.3 GHz 24 C/48 T, 150 W, or higher number Intel® Xeon® Gold/Platinum CPU SKU	Required
Memory	Option 1: DRAM only configuration: 384 GB (12 x 32 GB DDR4 2666 MHz)	Required
	Option 2: DRAM only configuration: 384 GB (24 x 16 GB DDR4 2666 MHz)	
	Option 3: DRAM + Intel® Optane™ persistent memory DRAM: 192 GB (12 x 16 GB DDR4 2666 MHz)	
Intel® QAT	Intel® C620 Series Chipset integrated on base board Intel® C627/C628 Chipset, integrated with NUMA connectivity to each CPU or minimum 16 Peripheral Component Interconnect express (PCIe) lane connectivity to one CPU	Required
Intel® Optane™ Persistent Memory	Option 1: 1 TB (8x 128 GB Intel® Optane™ persistent memory in 2-2-1 Topology)	Recommended
	Option 2: 1.5 TB (12x 128 GB Intel® Optane™ persistent memory in 2-2-2 Topology)	
Network Adapter	Option 1: Dual Port 25 GbE Intel® Ethernet Network Adapter X710-DA4 SFP+, or	Required
	Option 2: Dual Port 100 GbE Intel® Ethernet Network Adapter E810-CQDA2 QSFP28	
Storage (Boot Drive)	Intel® SATA Solid State Drive D3 S4510 at 480 GB or equivalent boot drive	Required
Storage (Capacity)	Intel® NVMe SSD DC P4510 Series at 2 TB or equivalent (Recommended NUMA Aligned)	Required
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	Required
Additional Plug-in cards	N/A	

Table 8. Hardware Components for Host Plus – 3rd Generation Intel Xeon Scalable Processor

INGREDIENT	REQUIREMENT	REQUIRED/ RECOMMENDED
3rd Generation Intel Xeon Scalable processors	Intel® Xeon® Gold 6338N CPU @ 2.2 GHz 32 C/64 T, 185 W, or higher number Intel® Xeon® Gold or Platinum CPU SKU	Required
Memory	Option 1: DRAM only configuration: 512 GB (16x 32 GB DDR4, 2666 MHz)	Required
	Option 2: DRAM only configuration: 512 GB (32x 16 GB DDR4, 2666 MHz)	

INGREDIENT	REQUIREMENT	REQUIRED/ RECOMMENDED
Intel® QAT	Intel® C620 Series Chipset integrated on base board Intel® C627/C628 Chipset, integrated with NUMA connectivity to each CPU or minimum 16 Peripheral Component Interconnect express (PCIe) lane connectivity to one CPU	Required
Intel® Optane™ Persistent Memory	Option 1: 1 TB (8x 128 GB Intel® Optane™ persistent memory in 8+4 Topology)	Recommended
	Option 2: 2 TB (16x 128 GB Intel® Optane™ persistent memory in 8+8 Topology)	
Network Adapter	Option 1: Intel® Ethernet Network Adapter E810-CQDA2	Required
	Option 2: Intel® Ethernet Network Adapter E810-2CQDA2	
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 4 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	Required
Additional Plug-in cards	Intel® Server Graphics 1 card	Optional

Table 9. Hardware Components for Host Plus – 4th Generation Intel Xeon Scalable Processor

INGREDIENT	REQUIREMENT	REQUIRED/ RECOMMENDED
4th Generation Intel Xeon Scalable processors	Intel® Xeon® Gold 6428N processor at 1.8GHz, 32 C/64 T, 185 W	Required
Memory	Option 1: DRAM only configuration: 512 GB (16x 32 GB DDR5)	Required
	Option 2: DRAM only configuration: 512 GB (32x 16 GB DDR5)	
Intel® QAT	Integrated in the processor	Required
Intel® Optane™ Persistent Memory	Option 1: 1 TB (8x 128 GB Intel® Optane™ persistent memory in 8+4 Topology)	Recommended
	Option 2: 2 TB (16x 128 GB Intel® Optane™ persistent memory in 8+8 Topology)	
Network Adapter	Option 1: Intel® Ethernet Network Adapter E810-CQDA2	Required
	Option 2: Intel® Ethernet Network Adapter E810-2CQDA2	
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 4 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	Required
Additional Plug-in cards	Intel® Server Graphics 1 card	Optional

3.3 Hardware BOMs for all VMRA Configuration Profiles

The following tables list the hardware BOMs for host base and plus.

The profiles for hosts vary with respect to network interface card, Intel® QuickAssist Technology, and BIOS profiles. You may choose based on the requirement for the workloads to be run in the cluster.

Table 10. Host Base Hardware Setup for all Configuration Profiles – 2nd Generation Intel Xeon Scalable Processor

NAME	Host_2ndGen_Base_1	Host_2ndGen_Base_2	Host_2ndGen_Base_3
Platform	S2600WFQ	S2600WFQ	S2600WFQ
CPU/node	2x 6230 or 6230N	2x 6230 or 6230N	2x 6230 or 6230N
Mem	384 GB	384 GB	384 GB
Intel Optane Persistent Memory	Recommended – 512 GB	Recommended – 512 GB	Recommended – 512 GB
Network Adapter	2x XXV710-DA2 or 2x E810-CQDA2	2x XXV710-DA2	2x XXV710-DA2
Storage (Boot Media)	Required - 2x	Required - 2x	Required - 2x

Storage (Capacity)	Required- 2x (1 per NUMA)	Required- 2x (1 per NUMA)	Required- 2x (1 per NUMA)
LOM	No	No	Yes
Intel® QAT	No	Optional	Yes
Additional Plug-in cards	No	No	No
BIOS Configuration			
Intel® HT Technology enabled	Yes	Yes	Yes
Intel® VT-x enabled	Yes	Yes	Yes
Intel® VT-d enabled	Yes	Yes	Yes
BIOS Profile	Energy Balance	Deterministic	Max Performance
Virtualization enabled	Yes	Yes	Yes

Table 11. Host Plus Hardware Setup for all Configuration Profiles – 2nd Generation Intel Xeon Scalable Processor

NAME	Host_2ndGen_Plus_1	Host_2ndGen_Plus_2
Platform	S2600WFQ	S2600WFQ
CPU/node	2x 6252 or 6252N	2x 6252 or 6252N
Mem	384 GB	384 GB
Intel Optane Persistent Memory	Recommended – 1 TB/1.5 TB	Recommended – 1 TB/1.5 TB
Network Adapter	2x E810-CQDA2	2x E810-CQDA2
Storage (Boot Media)	Required – 256 GB	Required – 256 GB
LOM	No	Yes
Intel® QAT	No	Yes
Additional Plug-in cards	No	No
BIOS Configuration		
Intel® HT Technology enabled	Yes	Yes
Intel® VT-x enabled	Yes	Yes
Intel® VT-d enabled	Yes	Yes
BIOS Profile	Deterministic	Max Performance
Virtualization enabled	Yes	Yes

Table 12. Host Base Hardware Setup for all Configuration Profiles – 3rd Generation Intel Xeon Scalable Processor

NAME	Host_3rdGen_Base_1	Host_3rdGen_Base_2	Host_3rdGen_Base_3
Platform	M50CYP	M50CYP	M50CYP
CPU/node	2x 5318N 24c	2x 5318N 24c	2x 5318N 24c
Mem	512 GB	512 GB	512 GB
Intel Optane Persistent Memory	Recommended – 512 GB	Recommended – 512 GB	Recommended – 512 GB
Network Adapter	2x E810-CQDA2	2x E810-CQDA2	2x E810-2CQDA2 or 4x E810-CQDA2
Storage (Boot Media)	Required - 2x	Required - 2x	Required - 2x
Storage (Capacity)	Required- 2x (1 per NUMA)	Required- 2x (1 per NUMA)	Required- 2x (1 per NUMA)
LOM	No	Yes	No
Intel® QAT	No	Yes	Optional
Additional Plug-in cards	No	No	No

BIOS Configuration			
Intel® HT Technology enabled	Yes	Yes	Yes
Intel® VT-x enabled	Yes	Yes	Yes
Intel® VT-d enabled	Yes	Yes	Yes
BIOS Profile	Energy Balance	Max Performance	Deterministic
Virtualization enabled	Yes	Yes	Yes

Table 13. Host Plus Hardware Setup for all Configuration Profiles – 3rd Generation Intel Xeon Scalable Processor

NAME	Host_3rdGen_Plus_1	Host_3rdGen_Plus_2	Host_3rdGen_Plus_3
Platform	M50CYP	M50CYP	M50CYP
CPU/node	2x 6338N 32c	2x 6338N 32c	2x 6338N 32c
Mem	512 GB	512 GB	512 GB
Intel Optane Persistent Memory	Recommended – 512 GB	Recommended – 512 GB	Recommended – 512 GB
Network Adapter	2x E810-2CQDA2 or 4x E810-CQDA2	2x E810-2CQDA2	2x E810-2CQDA2 or 4x E810-CQDA2
Storage (Boot Media)	Required - 2x	Required - 2x	Required - 2x
Storage (Capacity)	Required- 4x (2 per NUMA)	Required- 4x (2 per NUMA)	Required- 4x (2 per NUMA)
LOM	Yes	Yes	No
Intel® QAT	Yes	No	Optional
Additional Plug-in cards	No	Intel Server GPU	No
BIOS Configuration			
Intel® HT Technology enabled	Yes	Yes	Yes
Intel® VT-x enabled	Yes	Yes	Yes
Intel® VT-d enabled	Yes	Yes	Yes
BIOS Profile	Max Performance	Max Performance	Deterministic
Virtualization enabled	Yes	Yes	Yes

Table 14. Host Base Hardware Setup for all Configuration Profiles – 4th Generation Intel Xeon Scalable Processor

NAME	Host_4thGen_Base_1	Host_4thGen_Base_2
Platform	Archer City / Quanta - S6Q	Archer City / Quanta - S6Q
CPU/node	2x 5418N	2x 5418N
Mem	512 GB	512 GB
Intel Optane Persistent Memory	Recommended – 512 GB	Recommended – 512 GB
Network Adapter	2x E810-CQDA2	2x E810-CQDA2

Storage (Boot Media)	Required - 2x	Required - 2x
Storage (Capacity)	Required- 2x (1 per NUMA)	Required- 2x (1 per NUMA)
LOM	No	Yes
Intel® QAT	Integrated in the processor	Integrated in the processor
Additional Plug-in cards	No	No
BIOS Configuration		
Intel® HT Technology enabled	Yes	Yes
Intel® VT-x enabled	Yes	Yes
Intel® VT-d enabled	Yes	Yes
BIOS Profile	Energy Efficiency Turbo	Max Performance Turbo
Virtualization Enable	Yes	Yes

Table 15. Host Plus Hardware Setup for all Configuration Profiles – 4th Generation Intel Xeon Scalable Processor

NAME	Host_4thGen_Plus_1	Host_4thGen_Plus_2	Host_4thGen_Plus_3
Platform	Archer City / Quanta - S6Q	Archer City / Quanta - S6Q	Archer City / Quanta - S6Q
CPU/node	2x 6428N	2x 6428N	2x 6428N
Mem	512 GB	512 GB	512 GB
Intel Optane Persistent Memory	Recommended – 2 TB	Recommended – 1 TB	Recommended – 2 TB
Network Adapter	2x E810-2CQDA2 or 4x E810-CQDA2	2x E810-2CQDA2 or 8x E810 XXVAM-DA4	2x E810-2CQDA2 or 4x E810-CQDA2
Storage (Boot Media)	Required - 2x	Required - 2x	Required - 2x
Storage (Capacity)	Required- 4x (2 per NUMA)	Required- 4x (2 per NUMA)	Required- 4x (2 per NUMA)
LOM	Yes	Yes	No
Intel® QAT	Integrated in the processor	Integrated in the processor	Integrated in the processor
Additional Plug-in cards	No	Intel Server GPU	No
BIOS Configuration			
Intel® HT Technology enabled	Yes	Yes	Yes
Intel® VT-x enabled	Yes	Yes	Yes
Intel® VT-d enabled	Yes	Yes	Yes
BIOS Profile	Energy Balance Turbo	Energy Balance Turbo	Max Performance Turbo
Virtualization enabled	Yes	Yes	Yes

3.4 Platform BIOS

This section provides BIOS Configuration Profiles for each of the VMRA Configuration Profiles. For details on how the BIOS configuration should be set per each Configuration Profile, go to tables in Section 3.3.

For more information about BIOS settings, visit https://www.intel.com/content/dam/support/us/en/documents/server-products/Intel_Xeon_Processor_Scalable_Family_BIOS_User_Guide.pdf.

Table 16. Platform BIOS Settings for 2nd Generation Intel® Xeon® Scalable Processor

MENU (ADVANCED)	PATH TO BIOS SETTING	BIOS SETTINGS	ENERGY BALANCE	MAX PERFORMANCE	DETERMINISTIC
Advanced	Processor Configuration	Intel® Hyper-Threading Tech	Enabled	Enabled	Enabled
		Intel® Virtualization Technology	Enabled	Enabled	Enabled
	Integrated IO Configuration	Intel® VT for Directed I/O	Enabled	Enabled	Enabled
Advanced/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
	CPU P-state control	Enhanced Intel SpeedStep® Technology	Enabled	Enabled	Disabled* [Read footnote]
		Activate PBF	Disabled	Enabled	Enabled
		Configure PBF	Disabled	Disabled	Disabled
		Intel® Turbo Boost Technology	Enabled	Enabled	Disabled* [Read footnote]
		Energy Efficient Turbo	Enabled	Disabled	N/A
		Intel Configurable TDP	Disabled	Disabled	Disabled
	Hardware P-states	Hardware P-states	Native Mode with no legacy Support	Disabled** [Read footnote]	Disabled** [Read footnote]
		EPP Enable	Enabled	Enabled	Enabled
		RAPL Prioritization	Disabled	Enabled	Enabled
	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
Advanced	Memory Configuration	IMC Interleaving	2-way interleave	2-way Interleave	2-way Interleave
	System Acoustic and Performance Configuration	Set Fan Profile	Acoustic	Performance	Performance
GPU	GPU Fz	Lock 900Mhz	Optional	Optional	Optional

* Enabled in the case where Intel® SST-BF is enabled to allow for configuration of individual core speeds.

** "Native Mode with No Legacy Support" where Intel® SST-BF need to be enabled

Table 17. Platform BIOS Settings for 3rd Generation Intel® Xeon® Scalable Processor

MENU (Advanced)	Path to BIOS Setting	BIOS Setting	Energy Balance	Max Performance with Turbo	Deterministic
Socket Configuration	Processor Configuration	Hyper-Threading	Enable	Enable	Enable
		XAPIC	Enable	Enable	Enable

		VMX	Enable	Enable	Enable
		Uncore frequency scaling	Enable	Enable	Disable
		Uncore frequency	800-2400	800-2400	2400
Power Configuration	Power and Performance	CPU Power and Performance Policy	Balance Performance	Performance	Performance
		Workload Configuration	I/O sensitive	I/O sensitive	I/O sensitive
	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	Disable	Disable	Disable
		AVX ICCP Pre Grant Level	NA	NA	NA
		AVX P1	Nominal	Nominal	Nominal
		Energy Efficient Turbo	Enable	Enable	Disable
		WFR Uncore GV rate Reduction	Enable	Enable	Enable
		GPSS timer	500us	0us	0us
		Intel Turbo Boost Technology	Enable	Enable	Disable
		Intel SpeedStep® Technology (P-states)	Enable	Enable	Disable
	Frequency Prioritization	RAPL Prioritization	Enable	Disable	Disable
	Hardware PM State Control	Hardware P-States	Native Mode with no legacy Support	Disable	Disable
		EPP enable	Enable	Disable	Disable
	CPU C State Control	Enable Monitor Mwait	Enable	Enable	Enable
		CPU C1 Auto Demotion	Enable	Disable	Disable
		CPU C1 Auto unDemotion	Enable	Disable	Disable
		CPU C6 Report	Enable	Enable	Disable
		Processor C6	Enable	Enable	Disable
		Enhanced Halt State (C1E)	Enable	Enable	Disable
		OS ACPI Cx	ACPI C2	ACPI C2	ACPI C2
	Energy Performance Bias	Power Performance Tuning	OS Controls EPB	OS Controls EPB	OS Controls EPB
		ENERGY_PERF_BIAS_CFG mode	Performance	Performance	Performance
		Workload Configuration	I/O Sensitive	I/O Sensitive	I/O Sensitive
	Package C State Control	Package C State	C6 Retention	C6 Retention	C0/C1 State
		Dynamic L1	Enable	Disable	Disable
		Package C-state Latency Negotiation	Disable	Disable	Disable
		PKGC_SA_PS_CRITERIA	Disable	Disable	Disable
Memory Configuration		Memory Configuration	2-way interleave	2-way interleave	2-way interleave
		Enforce POR	Enable	Enable	Enable
Platform Configuration	Miscellaneous Configuration	Serial Debug Message Level	Minimum	Minimum	Minimum

	PCI Express* Configuration	PCIe* ASPM Support	Per Port	Per Port	Per Port
	PCI Express* Configuration	PCIe* ASPM	Enable	Disable	Disable
	PCI Express* Configuration	ECRC generation and checking	Enable	Enable	Enable
Server Management		Resume on AC Power Loss	Power On	Power On	Power On
System Acoustic and Performance Configuration		Set Fan Profile	Acoustic	Performance	Performance

Table 18. Platform BIOS Settings for 4th Generation Intel® Xeon® Scalable Processor

MENU (ADVANCED)	PATH TO BIOS SETTING	BIOS SETTING	MAX PERFORMANCE WITH TURBO	ENERGY BALANCE TURBO
Socket Configuration	Processor Configuration	Hyper-Threading	Enable	Enable
		X2APIC	Enable	Enable
		VMX	Enable	Enable
		LLC Prefetch	Enable	Enable
		SNC	Disable	Disable
		Uncore RAPL	Disable	Enable
		Uncore frequency scaling	Disable	Enable
		Uncore frequency	1.6MHz (hex 0x10)	800MHz to 2.5GHz
Power Configuration	CPU P-state Control	EIST PSD Function	HW_ALL	HW_ALL
		Boot Performance Mode	Max. Performance	Max. Performance
		AVX License Pre-Grant	Disable	Disable
		AVX ICCP Pre Grant Level	NA	NA
		AVX P1 (ConfigTDP)	Nominal (default)	Nominal
		Energy Efficient Turbo	Disable	Enable
		GPSS timer	0us	0us
		Turbo	Enable	Enable
		Intel® SpeedStep®(Pstates) Technology	Enable	Enable
	Frequency Prioritization	RAPL Prioritization	Disable	Disable
	Common Ref Code	UMA-Based Clustering	Quadrant	Quadrant
	Hardware PM State Control	Hardware P-states	Native with no Legacy Support	Native with no Legacy Support
		EPP enable	Disable	Disable
	CPU C-state Control	Enable Monitor Mwait	Enable	Enable
		CPU C1 Auto Demotion	Disable	Disable
		CPU C1 Auto unDemotion	Disable	Disable
		Processor C6 or CPU C6 Report	Enable	Enable

		Enhanced Halt State (C1E)	Enable	Enable
		OS ACPI Cx	ACPI C2	ACPI C2
	Energy Performance Bias	Power Performance Tuning	OS Controls EPB	OS Controls EPB
		Workload Configuration	I/O Sensitive	Balanced
	Package C-state Control	Package C-state	C0/C1 State	C0/C1 State
		Dynamic L1	Disable	Disable
Memory Configuration		Memory Configuration	8-way interleave	8-way interleave
		Enforce POR / Memory Patrol Scrub	Enable/Enable	Enable/Enable
		Memory DIMM Refresh Rate	1x	2x
Platform Configuration	Miscellaneous Configuration	Serial Debug Message Level	Minimum	Minimum
	PCI Express* Configuration	PCIe* ASPM	Enable	Enable
		ECRC generation and checking	Enable	Enable
Server Management		Resume on AC Power Loss	Power On	Power On
System Acoustic and Performance Configuration		Set Fan Profile	Acoustic	Acoustic

4 Reference Architecture Software Components

Table 18 lists all the software components automatically deployed for generating VMRA Flavors per Configuration Profile specifications and their sources.

Table 18. Software Components

SOFTWARE FUNCTION	SOFTWARE COMPONENT	LOCATION
OS	Ubuntu 20.04 Kernel version: 5.4.0-26-generic Ubuntu 22.04 Kernel version: 5.15.0-25-generic	https://www.ubuntu.com
Data Plane Development Kit	DPDK 22.03	https://core.dpdk.org/download/
Open vSwitch with DPDK	OVS-DPDK v2.17.1	https://github.com/openvswitch/ovs
Vector Packet Processing	VPP 21.10	https://docs.fd.io/vpp/
Telegraf	1.1	https://github.com/intel/observability-telegraf
CollectD	a7cea43d9d2f67c38fbf0407786edbe660ee9072945f7bb272b55fd255e8eaca	https://www.collectd.org/
Grafana	8.5.3	https://www.grafana.com/
Prometheus	2.35.0	https://quay.io/prometheus/prometheus
Ansible	Ansible 4.10.0	https://www.ansible.com/
VMRA Ansible Playbook	v22.05	https://github.com/intel/container-experience-kits
Python	Python 3.8.x for Ubuntu 20.04 Python 3.10.4 for Ubuntu 22.04	https://www.python.org/
Kubespray	2.18	https://github.com/kubernetes-sigs/kubespray
Docker	20.10	https://www.docker.com/
Containerd	1.5.11	Dependency of other software - not downloaded independently
CRI-O	1.22	Dependency of other software - not downloaded independently
Container orchestration engine	Kubernetes v1.23.4	https://github.com/kubernetes/kubernetes
	Kubernetes v1.22.3	
	Kubernetes v1.21.5	
CPU Manager (native to Kubernetes)	Available natively in Kubernetes	N/A
Platform Aware Scheduling (TAS)	TAS 0.2	https://github.com/intel/platform-aware-scheduling
Platform Aware Scheduling (GAS)	GAS 0.3	https://github.com/intel/platform-aware-scheduling
k8s-prometheus-adapter	0.8.4	Dependency of other software - not downloaded independently
Kubernetes node-exporter	1.3.1	Dependency of other software - not downloaded independently
Kubernetes prometheus-operator	0.50.0	https://github.com/prometheus-operator/kube-prometheus
Kubernetes kube-rbac-proxy	0.11.0	Dependency of other software - not downloaded independently
Node Feature Discovery	0.11.0	https://github.com/kubernetes-sigs/node-feature-discovery
Multus CNI	3.8	https://github.com/intel/multus-cni
SR-IOV CNI	2.6.2	https://github.com/intel/sriov-cni
SR-IOV network device plugin	3.4.0	https://github.com/intel/sriov-network-device-plugin
SR-IOV Network Operator	3f1b2e2d0792e96f2c6031a8bc8e42a4a093e573844f15f6f6f2c086646b19ba	https://github.com/k8snetworkplumbingwg/sriov-network-operator

SOFTWARE FUNCTION	SOFTWARE COMPONENT	LOCATION
Device Plugins Operator	0.23	https://github.com/intel/intel-device-plugins-for-kubernetes
QAT device plugin	0.23	https://github.com/intel/intel-device-plugins-for-kubernetes
GPU device plugin	0.23	https://github.com/intel/intel-device-plugins-for-kubernetes
Intel® SGX device plugin	0.23	https://github.com/intel/intel-device-plugins-for-kubernetes
DSA device plugin	0.23	https://github.com/intel/intel-device-plugins-for-kubernetes
Userspace CNI	1.3	https://github.com/intel/userspace-cni-network-plugin
Bond CNI plugin	1.0	https://github.com/intel/bond-cni
Intel® Ethernet Drivers	i40e v2.19.3 ice v1.8.8 iavf v4.4.2.1	https://sourceforge.net/projects/e1000/files/i40e%20stable/2.19.3/ https://sourceforge.net/projects/e1000/files/ice%20stable/1.8.8/ https://sourceforge.net/projects/e1000/files/iavf%20stable/4.4.2.1/
Intel® Ethernet NVM Update Package 700 Series	8.70	https://www.intel.com/content/www/us/en/download/18190/non-volatile-memory-nvm-update-utility-for-intel-ethernet-network-adapter-700-series.html
Intel® Ethernet NVM Update Package 800 Series	3.20	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	Dynamic Device Personalization for Intel® Ethernet 700 Series Version 25.4	https://downloadmirror.intel.com/27587/eng/gtp.zip https://downloadmirror.intel.com/28940/eng/mplsogreudp.zip https://downloadmirror.intel.com/28040/eng/ppp-oe-ol2tpv2.zip https://downloadmirror.intel.com/29446/eng/esp-ah.zip https://downloadmirror.intel.com/29780/eng/ecpri.zip
	Intel® Ethernet 800 Series DDP specific package for Comms 1.3.35.0	https://www.intel.com/content/www/us/en/download/19660/intel-ethernet-800-series-telecommunication-comms-dynamic-device-personalization-ddp-package.html
Intel® QAT Drivers	QAT.L.4.18.0-0008	Dependency of other software - not downloaded independently
OpenSSL	openssl-3.0.3	https://github.com/openssl/openssl
		https://www.openssl.org/source/
OpenSSL QAT Engine	0.6.12	https://github.com/intel/QAT_Engine
Intel ipsec-mb	1.2	https://github.com/intel/intel-ipsec-mb
Istio Operator	1.13.1	https://github.com/istio/istio/releases/download/
istio-intel/trusted-certificate-issuer	0.1.0	https://hub.docker.com/r/intel/trusted-certificate-issuer
istio-intel/tcpip-bypass-ebpf	1.0	https://hub.docker.com/r/intel/istio-tcpip-bypass
Intel® RDT	4.3.0	Dependency of other software - not downloaded independently
QEMU (Ubuntu 22.04, 20.04)	Latest (6.0.2, 4.2.1)	https://www.qemu.org/
Libvirt	Latest (8.0.0, 6.0.0)	https://libvirt.org/
Virt-manager	Latest (4.0.0, 2.2.1)	https://virt-manager.org/

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. Details for each of the Configuration Profiles are described in Sections 7 through 12.

The VMs can be accessed from the Ansible Host. Start by changing to the root user. If the name of the VMs has not been changed, they can be accessed directly through SSH:

```
$ ssh vm-ctrl-1
$ ssh vm-work-1
```

Note: If different VM names have been specified, the above commands should use the updated names.

In the following sections, whenever "kubectl" is used it is assumed that you are connected to one of the controller nodes.

Verification guidelines and output examples can be found on GitHub, as listed in Table 19.

Table 19. Links to Verification Guidelines on GitHub

VERIFICATION STEP
Check the Kubernetes Cluster
Check DDP Profiles on Intel® Ethernet 700 and 800 Series Network Adapters
Check Node Feature Discovery
Check Topology Manager
Check SR-IOV Device Plugin
Check QAT Device Plugin
Check Multus CNI Plugin
Check SR-IOV CNI Plugin
Check Userspace CNI Plugin
Check Telemetry Aware Scheduling
Check Intel QAT Engine with OpenSSL

5.1 Check Grafana Telemetry Visualization

VMRA 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 CA and it is available in /etc/kubernetes/ssl/ca.crt

As the VMs use an internal network, port forwarding must be configured before Grafana is accessible. From the Ansible host, as the root user, run the following command to set up forwarding:

```
$ ssh -L <Ansible Host IP>:30000:localhost:30000 vm-ctrl-1
```

Note: If the VM names have been changed, replace “vm-ctrl-1” with the updated name.

Note: If there are additional jumps between your machine and the Ansible Host, it might be necessary to configure additional forwarding or proxies. These steps will depend on your local setup.

Visit Grafana at https://<Ansible Host IP>:30000/

VMRA comes with a set of dashboards from the kube-prometheus project (https://github.com/prometheus-operator/kube-prometheus). Dashboards are available in the Dashboards -> Manage menu.

Part 2:
Building a VMRA Step-by-Step

6 VMRA Setup – Applicable for All Configuration Profiles

This section is relevant for generating VMRA Flavors based on their Configuration Profiles. It provides the prerequisites for a system setup and includes information that enables you to review BIOS prerequisites and software BOMs at a glance. The information is presented in multi-column tables to provide an easy way to compare and assess the differences between the VMRA Flavors that are available.

After setting up the Kubernetes system, refer to the specific section from the following list to build the Configuration Profile Flavor:

Section 7, VMRA Basic Configuration Profile Setup
Section 8, VMRA Full Configuration Profile Setup
Section 9, VMRA On-Premises Edge Configuration Profile Setup
Section 10, VMRA Remote Central Office-Forwarding Configuration Profile Setup
Section 11, VMRA Regional Data Center Configuration Profile Setup
Section 12, VMRA Build-Your-Own Configuration Profile Setup

Note: The taxonomy for the VMRA Configuration Profile settings is defined in Section 1.4.

Note: Not all features supported in Container Bare Metal Reference Architecture are supported in Virtual Machine Reference Architecture due to the nature of virtualization and abstraction of hardware.

6.1 Set Up an Ansible Host

VMRA Kubernetes clusters require an Ansible host that stores information about all managed remote nodes. In general, any machine running a recent Linux distribution can be used as Ansible host for any of the supported VMRA deployments, as long as it meets the following basic requirements:

- Network connectivity to the VM host server, including SSH
- Internet connection (using Proxy if necessary)
- Git utility installed
- Python 3 installed

Step-by-step instructions for building the Ansible host are provided below for the same list of operating systems that are supported for the VM host server (see Section 2.3.3):

6.1.1 RHEL 8 as Ansible Host

1. Install the Linux OS using any method supported by the vendor. If using the iso image, choose the Minimal iso version, or select the "Minimal Install" (Basic functionality) option under Software Selection.
2. Make the proper configuration during installation for the following key elements: Network (Ethernet) port(s) IP Address; Host Name, Proxies (if necessary), and Network Time Protocol (NTP).
3. After the installation completes and the machine reboots, log in as root and confirm that it has a valid IP address and can connect (ping) to the VM host.
4. Make sure that the http and https proxies are set, if necessary, for internet access. The configuration can be completed with the *export* command or by including the following lines in the /etc/environment file:

   ```
   http_proxy=http://proxy.example.com:1080
   https_proxy=http://proxy.example.com:1080
   ```

   Then, load the proxies configuration in the current environment:

   ```
   # source /etc/environment
   ```

5. Install Git:

   ```
   # yum install -y git
   ```

6. Install Python 3:

   ```
   # yum -y install python3
   ```

The Ansible host box is now ready to deploy the VMRA. Follow the instructions in Section 2.5.

6.1.2 Ubuntu 20.04 LTS as Ansible Host

1. Install the OS using any method supported by the vendor. Either the Desktop or Server distribution can be used. Select the “Minimal installation” option under “Updates and Other software”.
2. Follow steps 2, 3, and 4 as described above for RHEL.
3. Update the installation:

   ```
   # sudo apt update
   ```

4. Install SSH utilities:

```
# sudo apt install openssh-server
```

5. Install Git:

```
# sudo apt install -y git
```

6. Install Python 3-pip:

```
# sudo apt install -y python3-pip
```

The Ansible host box is now ready to deploy the VMRA. Follow the instructions in Section 2.5.

6.2 Set Up the VM Host - BIOS Prerequisites

This section is applicable for all **VMRA Configuration Profiles.**

Enter the UEFI or BIOS menu and update the configuration as shown in Table 20 and Table 21.

Note: The method for accessing the UEFI or BIOS menu is vendor-specific, for example: https://www.dell.com/support/article/us/en/04/sln167315/how-to-boot-into-the-bios-or-the-lifecycle-controller-on-your-poweredge-server?lang=en

Table 20. BIOS Prerequisites for VM Host for Basic, Full, and Build_Your_Own Configuration Profiles

PROFILES	BASIC CONFIGURATION PROFILE	FULL CONFIGURATION PROFILE	BUILD-YOUR-OWN CONFIGURATION PROFILE
Configuration			
BIOS Profile	Energy Balance	Max Performance	Any
Grub Command Line (values are set by Ansible)			
Isolcpus	Optional	Yes	Optional
Hugepages	Optional	Yes	Optional
P-state=disable	Optional	Yes, No-SST-BF	Optional
Limit C-state	Optional	Yes	Optional

Table 21. BIOS Prerequisites for VM Host for On-Premises Edge, Remote Central Office-Forwarding, and Regional Data Center Configuration Profiles

PROFILES	ON-PREMISES EDGE CONFIGURATION PROFILE	REMOTE CENTRAL OFFICE-FORWARDING CONFIGURATION PROFILE	REGIONAL DATA CENTER CONFIGURATION PROFILE
Configuration			
BIOS Profile	Max Performance	Deterministic / Energy Balance	Max Performance
Grub Command Line (values are set by Ansible)			
Isolcpus	Yes	Yes	Optional
Hugepages	Yes	Yes	Optional
P-state=disable	No	Yes, No-SST-BF	Optional
Limit C-state	No	Yes	Optional

The BIOS profile referenced in these tables consists of a number of configurations in the power management, thermal management, and configuration for Intel® platform technologies such as Intel® Virtualization Technology, Intel® Hyper-Threading Technology, Intel SpeedStep® technology, and Intel® Turbo Boost Technology.

The table provides three different BIOS profiles.

1. Energy Balance
2. Max Performance
3. Deterministic

The configuration and values set per each BIOS profile are defined in Table 16 and Table 17.

Note: The above values are the recommended configuration options on the Intel® S2600WFQ and Intel® M50CYP server boards. Some server boards may not provide the same options that are documented in this table. Vendors typically provide options for max performance configuration with virtualization.

6.3 Configuration Dictionary - Group Variables

Table 22 lists the parameters available as group variables with their type (for example, Boolean, string, URL, list, integer), possible values, and descriptions. The variables in **bold** must be updated to match the target environment. The variables with blue highlight must be updated according to your infrastructure. Refer to the section that describes your Configuration Profile to see the parameters enabled for that Configuration Profile.

Table 22. Configuration Dictionary – Group Variables

COMPONENT	COMPONENT PARAMETER	TYPE	VALUE	DESCRIPTION/COMMENT
Common Cluster Configuration				
Kubernetes		Boolean	true/false	Specifies whether to deploy Kubernetes
	kube_version	String	v1.23.4	Kubernetes version
	container_runtime	String	docker, crio, containerd	Container runtime to use as base engine for cluster deployment
	docker_version	String	20.10	Docker version
	containerd_version	String	1.5.11	Containerd version
	crio_version	String	1.22	CRI-O version
	update_all_packages	Boolean	false	Runs system-wide package update (apt dist-upgrade, yum update, ...). Tip: Can be set using host_vars for more granular control.
	http_proxy	**URL**	**http://proxy.example.com:1080**	**HTTP proxy address. Comment out if your cluster is not behind proxy.**
	https_proxy	**URL**	**http://proxy.example.com:1080**	**HTTPS proxy address. Comment out if your cluster is not behind proxy.**
	additional_no_proxy	**Comma-separated list of addresses**	**.example.com**	**Additional URLs that are not behind proxy, for example your corporate intra network DNS domain, e.g., ".intel.com". Note: Kubernetes nodes addresses, pod network, etc. are added to no_proxy automatically.**
	kube_network_plugin_multus	Boolean	True	Specifies whether to use the network plugin Multus
	multus_version	String	V3.8	Multus version
	kube_network_plugin	String	calico/flannel	Specifies networking CNI to use
	kube_pods_subnet	CIDR	10.244.0.0/16	Kubernetes pod subnet. Make sure that it matches your CNI plugin requirements (Calico by default) and doesn't overlap with your corporate LAN.
	kube_service_addresses	CIDR	10.233.0.0/18	Kubernetes service subnet. Make sure that it matches your CNI plugin requirements (Calico by default) and doesn't overlap with your corporate LAN.
	kube_proxy_mode	String	Iptables	Instructs kube_proxy how to set up NAT and load balancing functions
	kube_proxy_nodeport_addresses_cidr	CIDR	127.0.0.0/8	Kubernetes service subnet
	cluster_name	DNS domain	cluster.local	Name of the cluster
	registry_local_address	String	"localhost:30500"	Container registry address IP and port
	psp_enabled	Boolean	true/false	Enable pod security policy admission controller and create minimal set of rules
	always_pull_enabled	Boolean	true/false	Set image pull policy to Always. Pulls images before starting containers. Valid credentials must be configured.
Node Feature Discovery				
nfd_enabled		Boolean	true/false	Specifies whether to deploy Node Feature Discovery
	nfd_version	String	0.11	NFD version
	nfd_build_image_locally	Boolean	false	Builds NFD image locally instead of using the one from public registry.

COMPONENT	COMPONENT PARAMETER	TYPE	VALUE	DESCRIPTION/COMMENT
	nfd_namespace	String	kube-system	Kubernetes namespace used for NFD deployment
	nfd_sleep_interval	String	60s	Defines how often NFD queries node status and update node labels
Native Built-in Kubernetes CPU Manager				
native_cpu_manager_enabled		Boolean	true/false	Kubernetes CPU manager controls CPU management policies on the nodes. Setting this option as "true" enables the "static" policy; otherwise the default "none" policy is used.
	native_cpu_manager_system_reserved_cpus	Kubernetes millicores	2000m	Number of CPU cores to be reserved for housekeeping (2000m = 2000 millicores = 2 cores)
	native_cpu_manager_kube_reserved_cpus	Kubernetes millicores	1000m	Number of CPU cores to be reserved for Kubelet
	native_cpu_manager_reserved_cpus	Comma-separated list of integers or integer ranges	0,1,2	Explicit list of the CPUs reserved from pods scheduling. Supported only with kube_version 1.17 and newer, overrides 2 previous options.
Topology Manager (Kubernetes Built-in)[2]				
topology_manager_enabled		Boolean	true/false	Enables Kubernetes built-in Topology Manager
	topology_manager_policy	String, options: none, best-effort, restricted, single-numa-node	best-effort	Topology Manager policy
Intel SR-IOV Network Device Plugin				
sriov_network_operator_enabled		Boolean	true/false	Enables SR-IOV Network Operator
	sriov_network_operator_namespace	String	sriov-network-operator	Kubernetes namespace used to deploy SR-IOV network operator
sriov_net_dp_enabled		Boolean	true/false	Enables SR-IOV network device plugin
	sriov_net_dp_namespace	String	kube-system	Kubernetes namespace used to deploy SR-IOV network device plugin
	sriov_net_dp_build_image_locally	Boolean	true/false	Build and store image locally or use one from public external registry
	sriovdp_config_data	Multi-line string in JSON format	Two resource pools for kernel stack and DPDK-based networking respectively	SR-IOV network device plugin configuration. For more information on supported configurations, refer to https://github.com/intel/sriov-network-device-plugin#configurations
Intel Device Plugins for Kubernetes				
Intel_dp_namespace		String	kube-system	Kubernetes namespace used to deploy Intel device plugin operator
dsa_dp_enabled		Boolean	true/false	Enables Intel DSA device plugin
	configure_dsa_devices	Boolean	true/false	Specifies whether to configure DSA devices
	dsa_devices	List	[ ]	DSA devices to configure worker queues for
qat_dp_enabled		Boolean	true/false	Enables Intel QAT device plugin
	qat_dp_namespace	String	kube-system	Namespace used for Intel QAT device plugin

[2] Workloads and configurations. Results may vary.

COMPONENT	COMPONENT PARAMETER	TYPE	VALUE	DESCRIPTION/COMMENT
sgx_dp_enabled		Boolean	true/false	Enables Intel SGX device plugin
	sgx_dp_build_image_locally	Boolean	true/false	Build and store image locally or use one from public external registry
	sgx_aesmd_namespace	String	kube-system	Kubernetes namespace used to deploy SGX device plugin
	sgx_dp_provision_limit	Integer	20	
	sgx_dp_enclave_limit	Integer	20	
gpu_dp_enabled		Boolean	true	Enables Intel GPU device plugin
	gpu_dp_namespace	String	kube-system	Namespace used for Intel GPU device plugin
Service Mesh				
Istio_enabled		Boolean	true/false	Enables Istio service mesh for Kubernetes
Intel Telemetry Aware Scheduling				
tas_enabled		Boolean	true/false	Enables Intel Telemetry Aware Scheduling
	tas_namespace	String	monitoring	Kubernetes namespace used for TAS deployment
	tas_enable_demo_policy	Boolean	false	Creates demo TAS policy
Telemetry Configuration				
collectd_enabled		Boolean	true/false	Gather platform metrics with collectd
	collectd_scrap_interval	Integer	30	Duration to gather metrics using collectd
telegraf_enabled		Boolean	true/false	Gather platform metrics with telegraf
	telegraf_scrap_interval	Integer	30	Duration to gather metrics using Telegraf
Example Network Attachment Definitions (Ready to Use Examples of Custom CNI Plugin Configuration)				
example_net_attach_defs		List of dictionaries	[ ]	Example network attached definition objects to create
	userspace_ovs_dpdk	Boolean	true/false	Example net-attach-def for Userspace CNI with OVS-DPDK
	userspace_vpp	Boolean	true/false	Example net-attach-def for Userspace CNI with VPP
	sriov_net_dp	Boolean	true/false	Example net-attach-def for SR-IOV Net DP and SR-IOV CNI

6.4 Configuration Dictionary - Host Variables

Table 23 lists the parameters available as host variables with their type (for example, Boolean, string, URL, list, integer), possible values, and descriptions. The variables in **bold** must be updated to match the target environment. The variables with blue highlight must be updated according to your infrastructure. Refer to the section that describes your Configuration Profile to see the parameters enabled for that Configuration Profile.

Table 23. Configuration Dictionary - Host Variables for VMRA

COMPONENT	COMPONENT PARAMETER	TYPE	VALUE	DESCRIPTION/COMMENT
VM Configuration				
host		**List of dictionaries**	**n/a**	**Configuration of the host on which VMs will be deployed for VMRA**
	host[1].dhcp	**Integer**	**120**	**The vxLan identifier for the DHCP server supplying IP addresses to vm's on the vxLan. There must only be 1 DHCP server so this should only be specified on the first VM host.**

COMPONENT	COMPONENT PARAMETER	TYPE	VALUE	DESCRIPTION/COMMENT
	host[1].vxlan_gw_ip	String	“40.0.0.1/24”	Vxlan gateway IP, defines IP subnet used by DHCP. IP addresses from that subnet are assigned to all VMs.
	host[1].cpu_host_os	Integer	16	Number of CPU’s reserved for the host OS and not automatically assigned to VM’s. Defaults to 16.
	host[1].vm_image_distribution	String	ubuntu	Selected OS for VM’s. Supported values [‘ubuntu’, ‘rocky’]
	host[1].vm_image_version_ubuntu	String	“22.04”	Selected version if the VM OS is ubuntu. Supported values ['20.04', '21.10', '22.04']
	host[1].vm_image_version_rocky	String	“8.5”	Selected version if the VM OS is rocky. Supported values ['8.5']
	host[*].vxlan_device	String	“eno2”	The interface, which should be used for the vxLAN overlay network. On some systems, this might be eth1 or use the system naming convention, e.g., enp0s10
	host[*].vm_hashed_passwd	String	“xxxxxxxxx”	Parameter is used to configure password for root user inside VMs. Default value is just placeholder, which needs to be changed to real hashed password before deployment. To create hashed password with SHA512 use e.g.: openssl passwd -6 -salt SaltSalt <your_password>
vm’s		List of dictionaries	n/a	Configuration of VMs to be deployed for VMRA
	vms[*].type	String, options: “ctrl”, “work”	“ctrl” for controller node, “work” for worker node	Defines the Kubernetes node type that the VM will be created for
	vms[*].name	String	“vm-ctrl-1”, “vm-work-1”	Hostname for the VM and the Kubernetes node
	vms[*].cpus	String	"8-11,64-67"	This can be uncommented and manually configured, but by default the CPU’s and NUMA node are allocated automatically. List of host cores assigned to the VM. Each VM must only request cores from one NUMA zone or socket
	vms[*].numa	Integer	0, 1	This can be uncommented and manually configured, but by default the CPU’s and NUMA node are allocated automatically. The NUMA zone or socket that the VM will be running on. Must be aligned with vms[*].cpus.
	vms[*].cpu_total	Integer	8, 16	The number of cores assigned to the VM. Must match the core count in vms[*].cpus. If set to 0, then all unallocated CPUs from the selected NUMA node will be allocated to the VM.
	vms[*].alloc_all	Boolean	true, false	If set to true, then cpu_total should be set to 0. This parameter will then allocate all unallocated CPUs from all NUMA nodes on the host to the associated VM.
	vms[*].memory	Integer	20480, 61440	Amount of RAM to assign the VM (in megabyte)
	vms[*].vxlan	Integer	120	The default vxLan identifier for the overlay network. This must be consistent across all Hosts and VM’s in the deployment.
	vms[*].pci	List of strings	"18:02.2", "18:02.3"	PCI devices (VFs) to assign the VM. This is only set for worker node VM’s

7 VMRA Basic Configuration Profile Setup

This section contains a step-by-step description of how to set up a Basic Configuration Profile.

To use the Basic Configuration Profile, perform the following steps:

1. Choose your hardware, set it up, and configure the BIOS. Refer to Section 7.1 for details.
2. Download the Ansible playbook for your Configuration Profile. Refer to Section 7.2 for details.
3. Set up the optional Ansible parameters using the information in the Configuration Profile tables. Refer to Section 7.3 for details.
4. Deploy the platform. Refer to Section 7.4 for details.
5. Validate the setup of your Kubernetes cluster. Refer to the tasks in Section 5 and run the validation processes according to the hardware and software components that you have installed.

Be aware of the definitions of terminology used in tables in this section (Table 24).

Table 24. Taxonomy for Basic Configuration Profile Setup

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 user's 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.

7.1 Step 1 - Set Up Basic Configuration Profile Hardware

This section describes the hardware BOM and the BIOS configuration recommendation for using the Basic Configuration Profile.

The tables in this section list the hardware BOM for the Basic Configuration Profile, including VM Host Base and Plus.

Table 25. Hardware Setup for Basic Configuration Profile – 2nd, 3rd, and 4th Generation Intel Xeon Scalable Processors

NODE OPTIONS	2ND GENERATION INTEL XEON SCALABLE PROCESSOR	3RD GENERATION INTEL XEON SCALABLE PROCESSOR	4TH GENERATION INTEL XEON SCALABLE PROCESSOR
VM Host Options	Host_2ndGen_Base_1	Host_3rdGen_Base_1	Host_4thGen_Base_1

7.2 Step 2 - Download Basic Configuration Profile Ansible Playbook

This section contains details for downloading the Basic Configuration Profile Ansible playbook. It also provides an overview of the Ansible playbook and lists the software that is automatically installed when the playbook is deployed.

Download the Basic Configuration Profile Ansible playbook using the steps described in Section 2.5.

7.2.1 Basic Configuration Profile Ansible Playbook Overview

The Ansible playbook for the Basic Configuration Profile allows you to provision a production-ready virtual infrastructure with or without a Kubernetes cluster. Every capability included in the Basic Configuration Profile playbook can be disabled or enabled. Refer to Figure 4 and the group and host variables tables below.

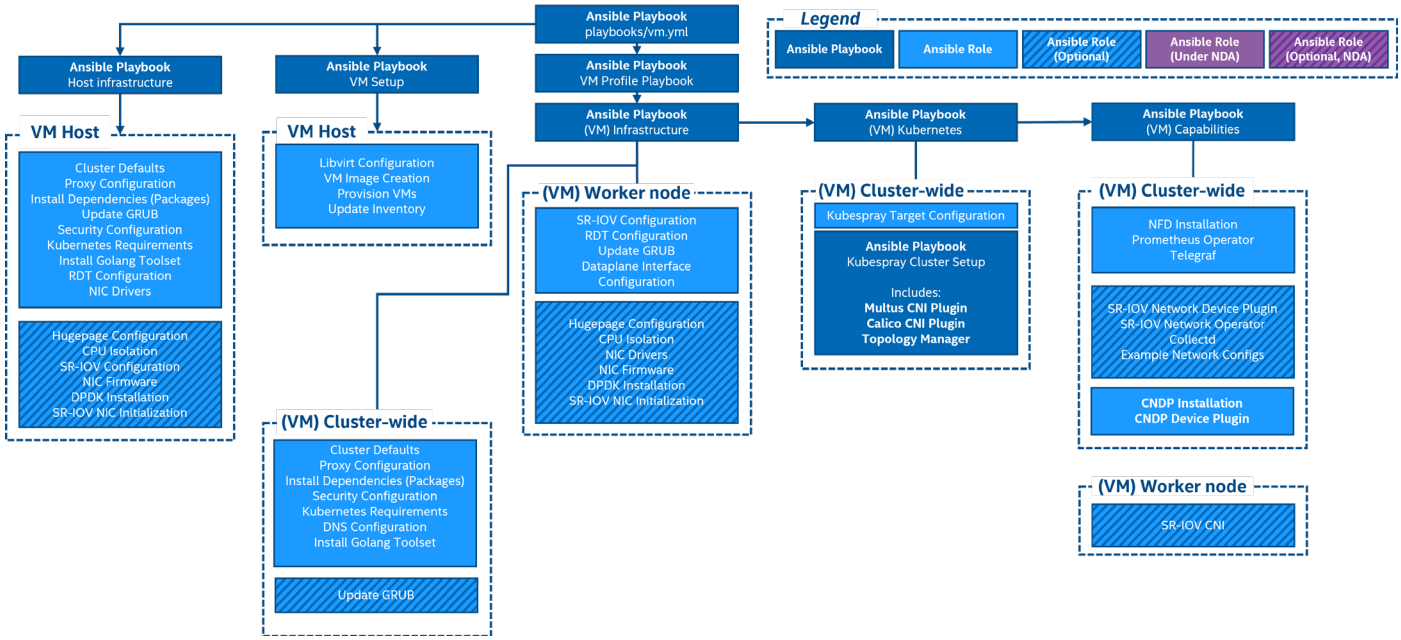

Figure 4. Basic Configuration Profile Ansible Playbook

7.3 Step 3 - Set Up Basic Configuration Profile

Review the optional Ansible group and host variables in this section and select options that match your desired configuration.

5. Update the inventory.ini file with your environment details as described in Section 2.5.3.
6. Create host_vars/<hostname>.yml for the target VM host server and all VMs of type "work" as specified in Section 2.5.4.
7. Update group and host variables to match your desired configuration as specified in Section 2.5.4.

Variables are grouped into two main categories:

1. Group variables – apply to both control and worker nodes and have cluster-wide impact.
2. Host variables – their scope is limited to a single worker node or the VM host.

The tables below are a summary of group and host variables. For lists showing all configurable properties, see Section 6.3 and Section 6.4. All of the variables are important but pay special attention to variables in **bold** as they almost always need to be updated to match the target environment.

7.3.1 Basic Configuration Profile Group Variables

Table 26. Basic Configuration Profile – Group Variables

COMPONENT	VALUE	
Kubernetes	true	For the list of all configurable properties, see Section 6.3
nfd_enabled	true	
topology_manager_enabled	true	
sriov_network_operator_enabled	false	
sriov_net_dp_enabled	false	
example_net_attach_defs, sriov_net_dp	false	

7.3.2 Basic Configuration Profile Host Variables[3]

Table 27. Basic Configuration Profile – Host Variables

COMPONENT	VALUE
iommu_enabled	false
sriov_cni_enabled	false

[3] Workloads and configurations. Results may vary.

COMPONENT	VALUE	
isolcpus_enabled	false	For the list of all configurable properties, see Section 6.4
dns_disable_stub_listener	true	

7.4 Step 4 - Deploy Basic Configuration Profile Platform

Note: You must download the Configuration Profile playbook as described in Section 7.2 and set it up as described in Section 7.3 before you complete this step.

In order to deploy the Basic Configuration Profile playbook, change the working directory to where you have cloned or unarchived the VMRA Ansible Playbook source code (as described in Section 2.5.2) and execute the command below:

```
$ ansible-playbook -i inventory.ini playbooks/vm.yml
```

7.5 Step 5 - Validate Basic Configuration Profile

Validate the setup of your Kubernetes cluster. Refer to the tasks in Section 5 and run the validation processes according to the hardware and software components that you have installed.

8 VMRA Full Configuration Profile Setup

This section contains a step-by-step description of how to set up a VMRA Flavor based Full Configuration Profile.

To use the VMRA Full Configuration Profile, perform the following steps:

1. Choose your hardware, set it up, and configure the BIOS. Refer to Section 8.1 for details.
2. Download the Ansible playbook for your Configuration Profile. Refer to Section 8.2 for details.
3. Configure the optional Ansible parameters using the information in the Configuration Profile tables. Refer to Section 8.3 for details.
4. Deploy the platform. Refer to Section 8.4 for details.
5. Validate the setup of your Kubernetes cluster. Refer to the tasks in Section 5 and run the validation processes according to the hardware and software components that you have installed.

Be aware of the definitions of terminology used in tables in this section (Table 28).

Table 28. Taxonomy for Full Configuration Profile Setup

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 user's 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.

8.1 Step 1 - Set Up Full Configuration Profile Hardware

This section describes the hardware BOM and the BIOS configuration recommendation for implementing a VMRA Flavor based on Full Configuration Profile.

The tables in this section list the hardware BOM for the Full Configuration Profile, including VM Host Base and Plus.

Table 29. Hardware Setup for Full Configuration Profile – 2nd, 3rd, and 4th Generation Intel Xeon Scalable Processors

NODE OPTIONS	2ND GENERATION INTEL XEON SCALABLE PROCESSOR	3RD GENERATION INTEL XEON SCALABLE PROCESSOR	4TH GENERATION INTEL XEON SCALABLE PROCESSOR
VM Host Options	Host_2ndGen_Plus_1	Host_3rdGen_Plus_1	Host_4thGen_Plus_1

8.2 Step 2 - Download Full Configuration Profile Ansible Playbook

This section contains details for downloading the Full Configuration Profile Ansible playbook. It also provides an overview of the Ansible playbook and lists the software that is automatically installed when the playbook is deployed.

Download the Full Configuration Profile Ansible playbook using the steps described in Section 2.5.

8.2.1 Full Configuration Profile Ansible Playbook Overview

The Ansible playbook for the Full Configuration Profile allows you to provision a production-ready virtual infrastructure with or without a a Kubernetes cluster. It also applies any additional requirements, such as host OS configuration or Network Adapter drivers and firmware updates. Full Configuration Profile playbook includes all features available through VMRA Ansible Playbook and provides one of the highest degrees of configurability. Every capability included in the Full Configuration Profile playbook can be disabled or enabled. To see which Ansible roles are included and executed by default, refer to Figure 5 and group and host variables tables below.

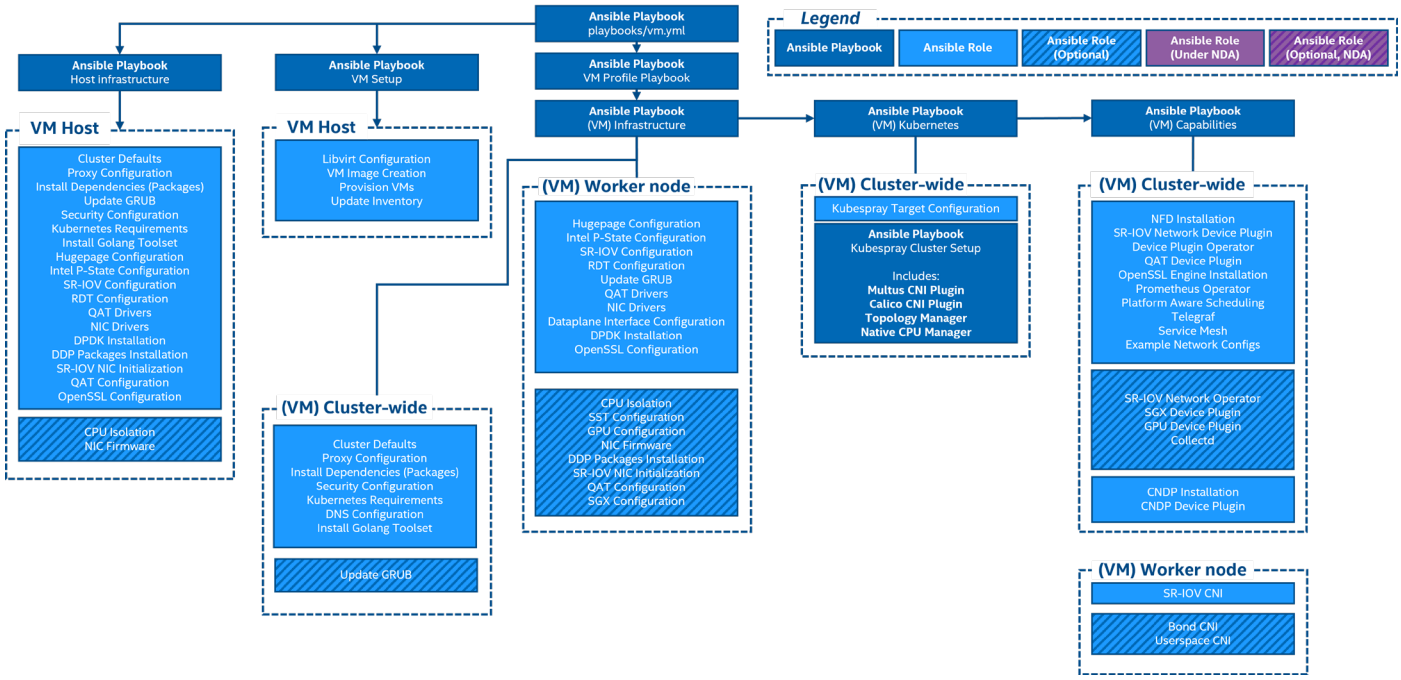

Figure 5. Full Configuration Profile Ansible Playbook

8.3 Step 3 - Set Up Full Configuration Profile

Review the optional Ansible group and host variables in this section and select options that match your desired configuration.

1. Update the inventory.ini file with your environment details as described in Section 2.5.3.
2. Create host_vars/<hostname>.yml for the target VM host server and all VMs of type "work" as specified in Section 2.5.4.
3. Update group and host variables to match your desired configuration as specified in Section 2.5.4.

Variables are grouped into two main categories:

1. Group variables – they apply to both control and worker nodes and have cluster-wide impact.
2. Host variables – their scope is limited to a single worker node or the VM host.

The tables below are a summary of group and host variables. For lists showing all configurable properties, see Section 6.3 and Section 6.4. All of the variables are important but pay special attention to variables in **bold** as they almost always need to be updated to match the target environment.

8.3.1 Full Configuration Profile Group Variables

Table 30. Full Configuration Profile – Group Variables

COMPONENT	VALUE	
Kubernetes	true	For the list of all configurable properties, see Section 6.3
nfd_enabled	true	
native_cpu_manager_enabled	true	
topology_manager_enabled	true	
sriov_network_operator_enabled	false	
sriov_net_dp_enabled	true	
sgx_dp_enabled	false	
gpu_dp_enabled	false	
qat_dp_enabled	true	
openssl_enabled	true	
kmra_enabled	false	
istio_enabled	true	
tas_enabled	true	
sst_pp_configuration_enabled	false	
example_net_attach_defs. userspace_ovs_dpdk	false	

8.3.2 Full Configuration Profile Host Variables

Table 31. Full Configuration Profile – Host Variables

COMPONENT	VALUE	
iommu_enabled	false	For the list of all configurable properties, see Section 6.4
sriov_cni_enabled	true	
bond_cni_enabled	false	
userspace_cni_enabled	false	
hugepages_enabled	true	
isolcpus_enabled	false	
dns_disable_stub_listener	true	
install_dpdk	true	
install_ddp_packages	false	
qat_devices	**[]**	

8.4 Step 4 - Deploy Full Configuration Profile Platform

Note: You must download the Configuration Profile playbook as described in Section 8.2 and configure it as described in Section 8.3 before you complete this step.

In order to deploy the Full Configuration Profile playbook, change the working directory to where you have cloned or unarchived the VMRA Ansible Playbook source code (as described in Section 2.5.2) and execute the command below:

```
$ ansible-playbook -i inventory.ini playbooks/vm.yml
```

8.5 Step 5 - Validate Full Configuration Profile

Validate the setup of your Kubernetes cluster. Refer to the tasks in Section 5 and run the validation processes according to the hardware and software components that you have installed.

9 VMRA On-Premises Edge Configuration Profile Setup

This section contains a step-by-step description of how to set up a VMRA based on On-Premises Edge Configuration Profile .

To use the VMRA On-Premises Edge Configuration Profile, perform the following steps:

1. Choose your hardware, set it up, and configure the BIOS. Refer to Section 9.1 for details.
   You also need to build your Kubernetes cluster.
2. Download the Ansible playbook for your Configuration Profile. Refer to Section 9.2 for details.
3. Configure the optional Ansible parameters using the information in the Configuration Profile tables. Refer to Section 9.3 for details.
4. Deploy the platform. Refer to Section 9.4 for details.
5. Validate the setup of your Kubernetes cluster. Refer to the tasks in Section 5 and run the validation processes according to the hardware and software components that you have installed.

Be aware of the definitions of terminology used in tables in this section (Table 32).

Table 32. Taxonomy for On-Premises Edge Configuration Profile Setup

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 user's 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.

9.1 Step 1 - Set Up On-Premises Edge Configuration Profile Hardware

This section describes the hardware BOM and the BIOS configuration recommendation for implementing VMRA based on On-Premises Edge Configuration Profile.

Table 33 lists the hardware BOM for the On-Premises Edge Configuration Profile, including VM host base and plus.

Table 33. Hardware Setup for On-Premises Edge Configuration Profile – 2nd, 3rd, and 4th Generation Intel Xeon Scalable Processors

NODE OPTIONS	2ND GENERATION INTEL XEON SCALABLE PROCESSOR	3RD GENERATION INTEL XEON SCALABLE PROCESSOR	4TH GENERATION INTEL XEON SCALABLE PROCESSOR
VM Host Options	Host_2ndGen_Base_2 or Host_2ndGen_Plus_1	Host_3rdGen_Base_2 or Host_3rdGen_Plus_1	Host_4thGen_Base_2 or Host_4thGen_Plus_1

9.2 Step 2 - Download On-Premises Edge Configuration Profile Ansible Playbook

This section contains details for downloading the On-Premises Edge Configuration Profile Ansible playbook. It also provides an overview of the Ansible playbook and lists the software that is automatically installed when the playbook is deployed.

Download the On-Premises Edge Configuration Profile Ansible playbook using the steps described in Section 2.5.

9.2.1 On-Premises Edge Configuration Profile Ansible Playbook Overview

The Ansible playbook for the On-Premises Edge Configuration Profile allows you to provision a production-ready virtual infrastructure with or without a Kubernetes cluster. It also applies any additional requirements, such as host OS configuration or Network Adapter drivers and firmware updates. Every capability included in the On-Premises Edge Configuration Profile playbook can be disabled or enabled. To see which Ansible roles are included and executed by default, refer to Figure 6 and group and host variables tables below.

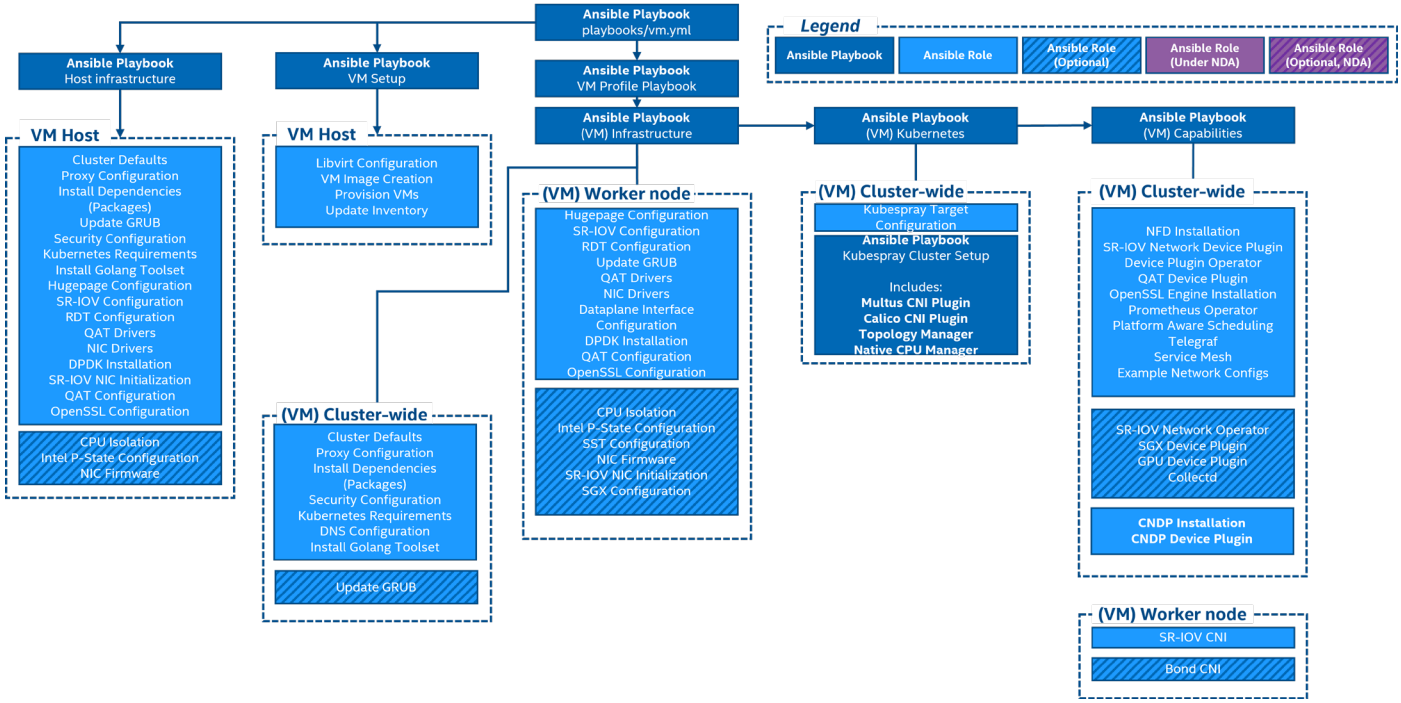

Figure 6. On-Premises Edge Configuration Profile Ansible Playbook

9.3 Step 3 - Set Up On-Premises Edge Configuration Profile

Review the optional Ansible group and host variables in this section and select options that match your desired configuration.

1. Update the inventory.ini file with your environment details as described in Section 2.5.3.
2. Create host_vars/<hostname>.yml for the target VM host server and all VMs of type "work" as specified in Section 2.5.4.
3. Update group and host variables to match your desired configuration as specified in Section 2.5.4.

Variables are grouped into two main categories:

1. Group variables – they apply to both control and worker nodes and have cluster-wide impact.
2. Host variables – their scope is limited to a single worker node or the VM host.

The tables below are a summary of group and host variables. For lists showing all configurable properties, see Section 6.3 and Section 6.4. All of the variables are important but pay special attention to variables in **bold** as they almost always need to be updated to match the target environment.

9.3.1 On-Premises Edge Configuration Profile Group Variables

Table 34. On-Premises Edge Configuration Profile – Group Variables

COMPONENT	VALUE	
Kubernetes	true	For the list of all configurable properties, see Section 6.3
nfd_enabled	true	
native_cpu_manager_enabled	true	
topology_manager_enabled	true	
sriov_network_operator_enabled	false	
sriov_net_dp_enabled	true	
sgx_dp_enabled	false	
gpu_dp_enabled	false	
qat_dp_enabled	true	
openssl_enabled	true	
kmra_enabled	false	
istio_enabled	true	
tas_enabled	true	
sst_pp_configuration_enabled	false	

COMPONENT	VALUE
example_net_attach_defs. userspace_ovs_dpdk	false

9.3.2 On-Premises Edge Configuration Profile Host Variables

Table 35. On-Premises Edge Configuration Profile – Host Variables

COMPONENT	VALUE	
iommu_enabled	false	For the list of all configurable properties, see Section 6.4
sriov_cni_enabled	true	
bond_cni_enabled	false	
hugepages_enabled	true	
isolcpus_enabled	false	
dns_disable_stub_listener	true	
install_dpdk	true	
qat_devices	**[]**	

9.4 Step 4 - Deploy On-Premises Edge Configuration Profile Platform

Note: You must download the Configuration Profile playbook as described in Section 9.2 and configure it as described in Section 9.3 before you complete this step.

In order to deploy the On-Premises Edge Configuration Profile playbook, change the working directory to where you have cloned or unarchived the VMRA Ansible Playbook source code (as described in Section 2.5.2) and execute the command below:

```
ansible-playbook -i inventory.ini playbooks/vm.yml
```

9.5 Step 5 - Validate On-Premises Edge Configuration Profile

Validate the setup of your Kubernetes cluster. Refer to the tasks in Section 5 and run the validation processes according to the hardware and software components that you have installed.

10 VMRA Remote Central Office-Forwarding Configuration Profile Setup

This section contains a step-by-step description of how to set up your VMRA Remote Central Office-Forwarding Configuration Profile Flavor.

To use the Remote Central Office-Forwarding Configuration Profile, perform the following steps:

1. Choose your hardware, set it up, and configure the BIOS. Refer to Section 10.1 for details. You also need to build your Kubernetes cluster.
2. Download the Ansible playbook for your Configuration Profile. Refer to Section 10.2 for details.
3. Configure the optional Ansible parameters using the information in the Configuration Profile tables. Refer to Section 10.3 for details.
4. Deploy the platform. Refer to Section 10.4 for details.
5. Validate the setup of your Kubernetes cluster. Refer to the tasks in Section 5 and run the validation processes according to the hardware and software components that you have installed.

Be aware of the definitions of terminology used in tables in this section (Table 36).

Table 36. Taxonomy for Remote Central Office-Forwarding

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 user's 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.

10.1 Step 1 - Set Up Remote Central Office-Forwarding Configuration Profile Hardware

This section describes the hardware BOM and the BIOS configuration recommendation for implementing a VMRA based Remote Central Office-Forwarding Configuration Profile.

The tables in this section list the hardware BOM for the Remote Central Office-Forwarding Configuration Profile, including VM host base and plus.

Table 37. Hardware Setup for Remote Central Office-Forwarding Configuration Profile – 2nd, 3rd, and 4th Generation Intel Xeon Scalable Processors

NODE OPTIONS	2ND GENERATION INTEL XEON SCALABLE PROCESSOR	3RD GENERATION INTEL XEON SCALABLE PROCESSOR	4TH GENERATION INTEL XEON SCALABLE PROCESSOR
VM Host Options	Host_2ndGen_Base_3 or Host_2ndGen_Plus_2	Host_3rdGen_Base_3 or Host_3rdGen_Plus_2	Host_4thGen_Base_3 or Host_4thGen_Plus_2

10.2 Step 2 - Download Remote Central Office-Forwarding Configuration Profile Ansible Playbook

This section contains details for downloading the Remote Central Office-Forwarding Configuration Profile Ansible playbook. It also provides an overview of the Ansible playbook and lists the software that is automatically installed when the playbook is deployed.

Download the Remote Central Office-Forwarding Configuration Profile Ansible playbook using the steps described in Section 2.5.

10.2.1 Remote Central Office-Forwarding Configuration Profile Ansible Playbook Overview

The Ansible playbook for the Remote Central Office-Forwarding Configuration Profile allows you to provision a production-ready virtual infrastructure with or without a Kubernetes cluster. It also applies any additional requirements, such as host OS configuration or network adapter drivers and firmware updates. Every capability included in the Remote Central Office-Forwarding Configuration Profile playbook can be disabled or enabled. To see which Ansible roles are included and executed by default, refer to Figure 7 and the group and host variables tables below.

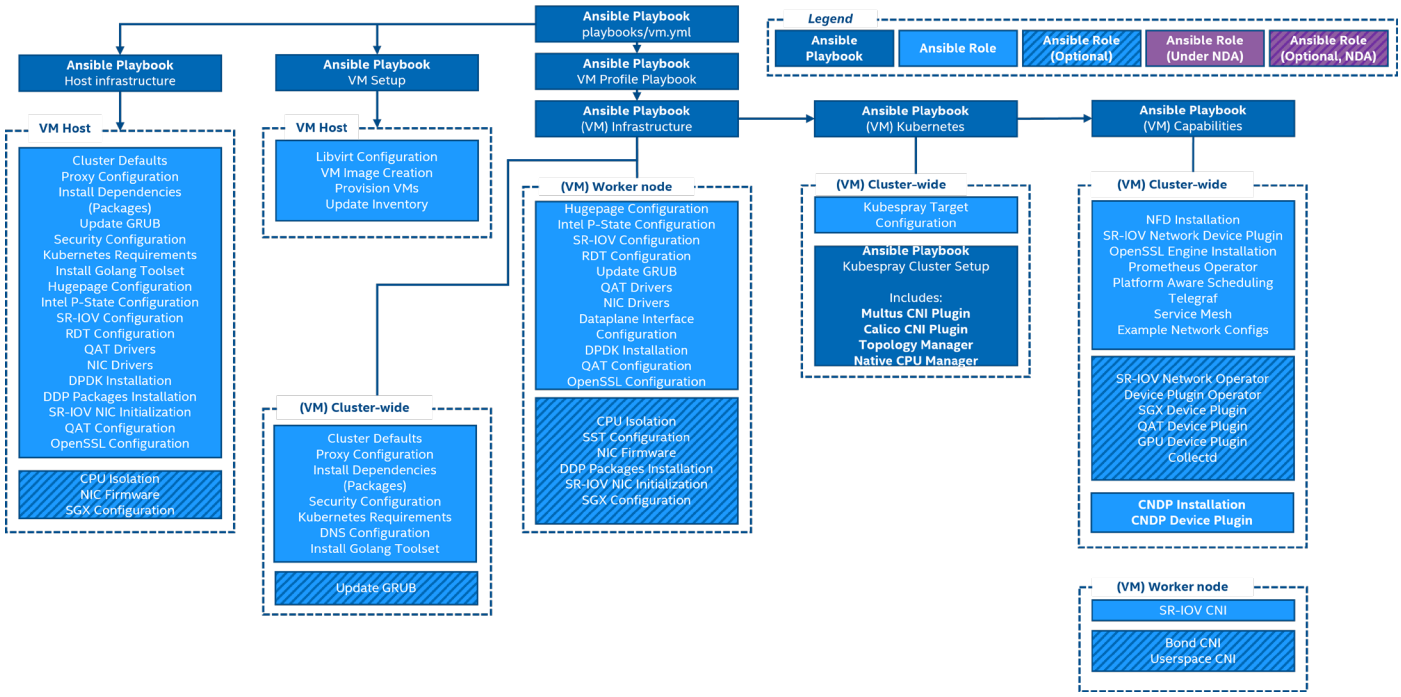

Figure 7. Remote Central Office-Forwarding Configuration Profile Ansible Playbook

10.3 Step 3 - Set Up Remote Central Office-Forwarding Configuration Profile

Review the optional Ansible group and host variables in this section and select options that match your desired configuration.

1. Update the inventory.ini file with your environment details as described in Section 2.5.3.
2. Create host_vars/<hostname>.yml for the target VM host server and all VMs of type "work" as specified in Section 2.5.4.
3. Update group and host variables to match your desired configuration as specified in Section 2.5.4.

Variables are grouped into two main categories:

1. Group variables – they apply to both control and worker nodes and have cluster-wide impact.
2. Host variables – their scope is limited to a single worker node or the VM host.

The tables below are a summary of group and host variables. For lists showing all configurable properties, see Section 6.3 and Section 6.4. All of the variables are important but pay special attention to variables in **bold** as they almost always need to be updated to match the target environment.

10.3.1 Remote Central Office-Forwarding Configuration Profile Group Variables

Table 38. Remote Central Office-Forwarding Configuration Profile – Group Variables

COMPONENT	VALUE	
Kubernetes	true	For the list of all configurable properties, see Section 6.3
nfd_enabled	true	
native_cpu_manager_enabled	true	
topology_manager_enabled	true	
sriov_network_operator_enabled	false	
sriov_net_dp_enabled	true	
sgx_dp_enabled	false	
gpu_dp_enabled	false	
qat_dp_enabled	false	
openssl_enabled	true	
kmra_enabled	false	
istio_enabled	true	
tas_enabled	true	
sst_cp_configuration_enabled	false	

COMPONENT	VALUE
example_net_attach_defs. userspace_ovs_dpdk	false

10.3.2 Remote Central Office-Forwarding Configuration Profile Host Variables

Table 39. Remote Central Office-Forwarding Configuration Profile – Host Variables

COMPONENT	VALUE	
iommu_enabled	false	For the list of all configurable properties, see Section 6.4
sriov_cni_enabled	true	
bond_cni_enabled	false	
userspace_cni_enabled	false	
hugepages_enabled	true	
isolcpus_enabled	false	
dns_disable_stub_listener	true	
install_dpdk	true	
install_ddp_packages	false	
qat_devices	**[]**	

10.4 Step 4 - Deploy Remote Central Office-Forwarding Configuration Profile Platform

Note: You must download the Configuration Profile playbook as described in Section 10.2 and configure it as described in Section 10.3 before you complete this step.

In order to deploy the Remote Central Office-Forwarding Configuration Profile playbook, change the working directory to where you have cloned or unarchived the VMRA Ansible Playbook source code (as described in Section 2.5.2)and execute the command below:

```
ansible-playbook -i inventory.ini playbooks/vm.yml
```

10.5 Step 5 - Validate Remote-Central Office Forwarding Configuration Profile

Validate the setup of your Kubernetes cluster. Refer to the tasks in Section 5 and run the validation processes according to the hardware and software components that you have installed.

11 VMRA Regional Data Center Configuration Profile Setup

This section contains a step-by-step description of how to set up a VMRA based Regional Data Center Configuration Profile .

To use the Regional Data Center Configuration Profile, perform the following steps:

1. Choose your hardware, set it up, and configure the BIOS. Refer to Section 11.1 for details. You also need to build your Kubernetes cluster.
2. Download the Ansible playbook for your Configuration Profile. Refer to Section 11.2 for details.
3. Configure the optional Ansible parameters using the information in the Configuration Profile tables. Refer to Section 11.3 for details.
4. Deploy the platform. Refer to Section 11.4 for details.
5. Validate the setup of your Kubernetes cluster. Refer to the tasks in Section 5 and run the validation processes according to the hardware and software components that you have installed.

Be aware of the definitions of terminology used in tables in this section (Table 40).

Table 40. Taxonomy for Regional Data Center Configuration Profile Setup

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 user's 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.

11.1 Step 1 - Set Up Regional Data Center Configuration Profile Hardware

This section describes the hardware BOM and the BIOS configuration recommendation for implementing a VMRA Flavor based Regional Data Center Configuration Profile .

The tables in this section list the hardware BOM for the Regional Data Center Configuration Profile, including VM host base and plus.

Table 41. Hardware Setup for Regional Data Center Configuration Profile – 2nd, 3rd, and 4th Generation Intel Xeon Scalable Processors

NODE OPTIONS	2ND GENERATION INTEL XEON SCALABLE PROCESSOR	3RD GENERATION INTEL XEON SCALABLE PROCESSOR	4TH GENERATION INTEL XEON SCALABLE PROCESSOR
VM Host Options	N/A*	Host_3rdGen_Plus_3	N/A*
*Configuration Profile only tested with 3rd Generation Intel Xeon Scalable processor			

11.2 Step 2 - Download Regional Data Center Configuration Profile Ansible Playbook

This section contains details for downloading the Regional Data Center Configuration Profile Ansible playbook. It also provides an overview of the Ansible playbook and lists the software that is automatically installed when the playbook is deployed.

Download the Regional Data Center Configuration Profile Ansible playbook using the steps described in Section 2.5.

11.2.1 Regional Data Center Configuration Profile Ansible Playbook Overview

The Ansible playbook for the Regional Data Center Configuration Profile allows you to provision a production-ready virtual infrastructure with or without a Kubernetes cluster. It also applies any additional requirements, such as host OS configuration or Network Adapter drivers and firmware updates. Every capability included in the Regional Data Center Configuration Profile playbook can be disabled or enabled. To see which Ansible roles are included and executed by default, refer to Figure 8 and the group and host variables tables below.

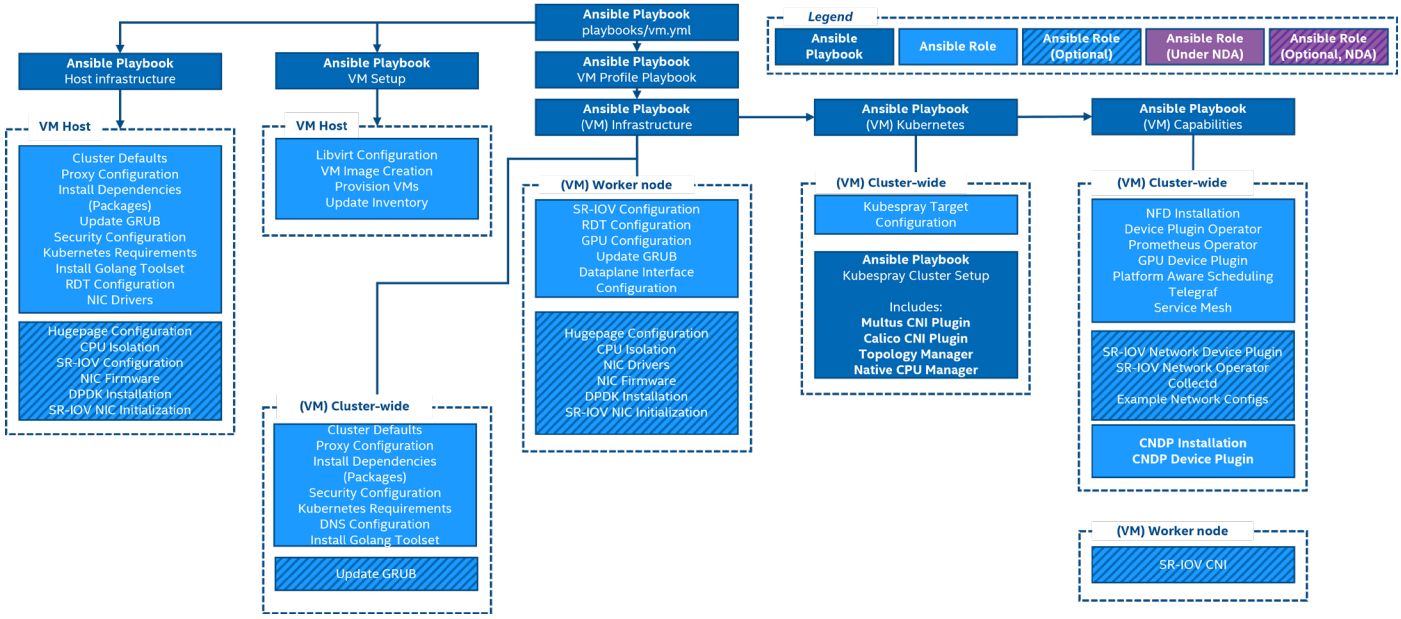

Figure 8. Regional Data Center Configuration Profile Ansible Playbook

11.3 Step 3 - Set Up Regional Data Center Configuration Profile

Review the optional Ansible group and host variables in this section and select options that match your desired configuration.

1. Update the inventory.ini file with your environment details as described in Section 2.5.3.
2. Create host_vars/<hostname>.yml for the target VM host server and all VMs of type "work" as specified in Section 2.5.4.
3. Update group and host variables to match your desired configuration as specified in Section 2.5.4.

Variables are grouped into two main categories:

1. Group variables – they apply to both control and worker nodes and have cluster-wide impact.
2. Host variables – their scope is limited to a single worker node or the VM host.

The tables below are a summary of group and host variables. For lists showing all configurable properties, see Section 6.3 and Section 6.4. All of the variables are important but pay special attention to variables in **bold** as they almost always need to be updated to match the target environment.

11.3.1 Regional Data Center Configuration Profile Group Variables

Table 42. Regional Data Center Configuration Profile – Group Variables

COMPONENT	VALUE	
Kubernetes	true	For the list of all configurable properties, see Section 6.3
nfd_enabled	true	
native_cpu_manager_enabled	true	
topology_manager_enabled	true	
sriov_network_operator_enabled	false	
sriov_net_dp_enabled	false	
gpu_dp_enabled	true	
Istio_enabled	true	
tas_enabled	true	
example_net_attach_defs. sriov_net_dp	false	

11.3.2 Regional Data Center Configuration Profile Host Variables

Table 43. Regional Data Center Configuration Profile – Host Variables

COMPONENT	VALUE
iommu_enabled	false
sriov_cni_enabled	false

COMPONENT	VALUE	
hugepages_enabled	false	For the list of all configurable properties, see Section 6.4
isolcpus_enabled	false	
dns_disable_stub_listener	true	
install_dpdk	false	

11.4 Step 4 - Deploy Regional Data Center Configuration Profile Platform

Note: You must download the Configuration Profile playbook as described in Section 11.2 and configure it as described in Section 11.3 before you complete this step.

In order to deploy the Regional Data Center Configuration Profile playbook, change the working directory to where you have cloned or unarchived the VMRA Ansible Playbook source code (as described in Section 2.5.2) and execute the command below:

```
ansible-playbook -i inventory.ini playbooks/vm.yml
```

11.5 Step 5 - Validate Regional Data Center Configuration Profile

Validate the setup of your Kubernetes cluster. Refer to the tasks in Section 5 and run the validation processes according to the hardware and software components that you have installed.

12 VMRA Build-Your-Own Configuration Profile Setup

This section contains a step-by-step description of how to set up a Build-Your-Own Configuration Profile.

To use the Build-Your-Own Configuration Profile, perform the following steps:

1. Choose your hardware, set it up, and configure the BIOS. Refer to Section 12.1 for details.
2. Download the Ansible playbook for your Configuration Profile. Refer to Section 12.2 for details.
3. Set up the optional Ansible parameters using the information in the Configuration Profile tables. Refer to Section 12.3 for details.
4. Deploy the platform. Refer to Section 12.4 for details.
5. Validate the setup of your Kubernetes cluster. Refer to the tasks in Section 5 and run the validation processes according to the hardware and software components that you have installed.

Be aware of the definitions of terminology used in tables in this section (Table 44).

Table 44. Taxonomy for Build-Your-Own Configuration Profile Setup

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 user's 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.

12.1 Step 1 - Set Up Build-Your-Own Configuration Profile Hardware

This section describes the hardware BOM and the BIOS configuration recommendation for implementing a VMRA based Build-Your-Own Configuration Profile .

The tables in this section list the hardware BOM for the Build-Your-Own Configuration Profile, including VM Host Base and Plus.

Table 45. Hardware Setup for Build-Your-Own Configuration Profile – 2nd, 3rd, and 4th Generation Intel Xeon Scalable Processors

NODE OPTIONS	2ND GENERATION INTEL XEON SCALABLE PROCESSOR	3RD GENERATION INTEL XEON SCALABLE PROCESSOR	4TH GENERATION INTEL XEON SCALABLE PROCESSOR
VM Host Options	Host_2ndGen_Base_1	Host_3rdGen_Base_1	Host_4thGen_Base_1

12.2 Step 2 - Download Build-Your-Own Configuration Profile Ansible Playbook

This section contains details for downloading the Build-Your-Own Configuration Profile Ansible playbook. It also provides an overview of the Ansible playbook and lists the software that is automatically installed when the playbook is deployed.

Download the Build-Your-Own Configuration Profile Ansible playbook using the steps described in Section 2.5.

12.2.1 Build-Your-Own Configuration Profile Ansible Playbook Overview

The Ansible playbook for the Build-Your-Own Configuration Profile allows you to provision a production-ready virtual infrastructure with or without a Kubernetes cluster. Every capability included in the Build-Your-Own Configuration Profile playbook can be disabled or enabled. Refer to Figure 9 and the group and host variables tables below.

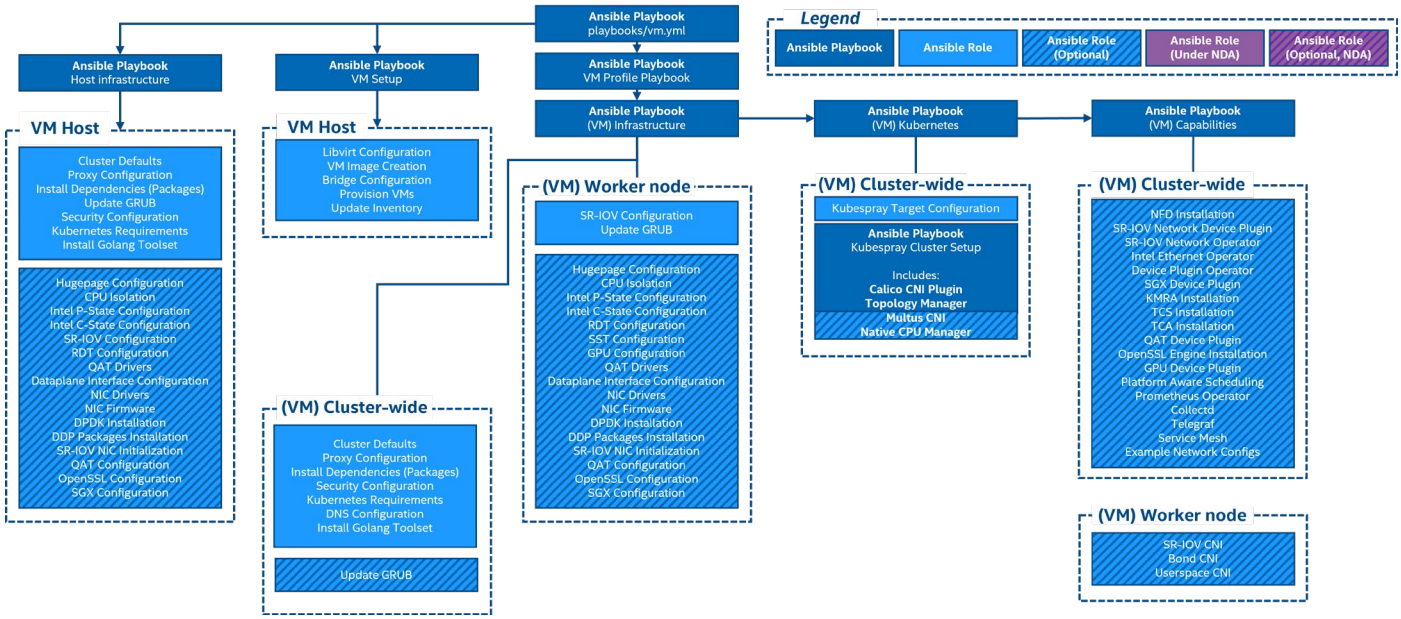

Figure 9. Build-Your-Own Configuration Profile Ansible Playbook

12.3 Step 3 - Set Up Build-Your-Own Configuration Profile

Review the optional Ansible group and host variables in this section and select options that match your desired configuration.

4. Update the inventory.ini file with your environment details as described in Section 2.5.3.
5. Create host_vars/<hostname>.yml for the target VM host server and all VMs of type "work" as specified in Section 2.5.4.
6. Update group and host variables to match your desired configuration as specified in Section 2.5.4.

Variables are grouped into two main categories:

1. Group variables – apply to both control and worker nodes and have cluster-wide impact.
2. Host variables – their scope is limited to a single worker node or the VM host.

The tables below are a summary of group and host variables. For lists showing all configurable properties, see Section 6.3 and Section 6.4. All of the variables are important but pay special attention to variables in **bold** as they almost always need to be updated to match the target environment.

12.3.1 Build-Your-Own Configuration Profile Group Variables

Table 46. Build-Your-Own Configuration Profile – Group Variables

COMPONENT	VALUE	
Kubernetes	true	For the list of all configurable properties, see Section 6.3
nfd_enabled	true	
topology_manager_enabled	true	
sriov_network_operator_enabled	false	
sriov_net_dp_enabled	false	
example_net_attach_defs, sriov_net_dp	false	

12.3.2 Build-Your-Own Configuration Profile Host Variables

Table 47. Build-Your-Own Configuration Profile – Host Variables

COMPONENT	VALUE	
iommu_enabled	false	For the list of all configurable properties, see Section 6.4
sriov_cni_enabled	false	
isolcpus_enabled	false	
dns_disable_stub_listener	true	

12.4 Step 4 - Deploy Build-Your-Own Configuration Profile Platform

Note: You must download the Configuration Profile playbook as described in Section 12.2 and set it up as described in Section 12.3 before you complete this step.

In order to deploy the Build-Your-Own Configuration Profile playbook, change the working directory to where you have cloned or unarchived the VMRA Ansible Playbook source code (as described in Section 2.5.2) and execute the command below:

```
$ ansible-playbook -i inventory.ini playbooks/vm.yml
```

12.5 Step 5 - Validate Build-Your-Own Configuration Profile

Validate the setup of your Kubernetes cluster. Refer to the tasks in Section 5 and run the validation processes according to the hardware and software components that you have installed.

Part 3:
Release Notes

Appendix A VMRA Release Notes

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

A.1 VMRA 22.05 New Features

This release of the VMRA supports deployment of Kubernetes across virtual machines on multiple hosts. For a multi-host deployment, a VXLan overlay network is created for intra-host communications, with VM's on each host connected via a VXLan bridge.

Automatic CPU pinning is supported to assign vCPU's to cores within the same NUMA zone.

This release supports the following operating systems on Hosts and VM Guests – Rocky 8.5, Ubuntu 21.10 and 22.04. Support for Ubuntu 20.04LTS is retained.

A.2 VMRA 22.01 New Features

This is the first release of VMRA.

A.3 Known Issues

Issue:

VFs specified in host_vars "dataplane_interfaces are not bound to the expected VF driver

Detail:

Due to VMs not having access to physical functions (PFs) on NICs attached to the VM Host, the configuration of VFs is skipped inside VMs. As a result, VFs will be bound to the Linux "iavf" driver and show up as "netdevice" devices through the SR-IOV Network Device Plugin.

Workaround:

Follow the steps listed in Table 19 (Check SR-IOV device plugin) to rebind VFs to the correct driver.

Issue:

VMRA 22.05 introduces AF_XDP and CNDP support for SR-IOV Virtual Functions (VFs) using the Linux kernel iavf driver. The iavf driver does not currently support XDP or AF_XDP zero-copy, so the kernel's generic eXpress Data Path (XDP) is used. This results in extra per-packet overhead due to allocation of SKBs and requires a copy to get the packet data from the kernel to the AF_XDP socket in user space.

Detail:

Applications using AF_XDP sockets on devices that do not support XDP or AF_XDP zero-copy will generally result in lower performance than applications using AF_XDP sockets on devices that support XDP and AF_XDP zero-copy.

Workaround:

AF_XDP in XDP_SKB mode is used for devices that do not support AF_XDP zero-copy.

Issue:

GPU Aware Scheduling (GAS) is enabled for the Regional Data Center profile, even though it is not supported or tested for VMRA.

Detail:

As part of Platform Aware Scheduling (PAS), the GPU Aware Scheduling (GAS) extender is enabled for the Regional Data Center profile. GPU virtualization is not currently supported in VMRA, which might cause unexpected behavior of the extender.

Workaround:

If configuring the Regional Data Center profile (regional_dc), manually update "gas_enabled" in group_vars/all.yml to "false"

[4] Workloads and configurations. Results may vary.

Issue:

QAT Devices are not providing additional performance for 2nd and 3rd generation through offloading.

Detail:

While QAT Devices can be configured and will show up in the Kubernetes cluster as an allocatable resource, they do not provide the expected performance increase. When testing with the OpenSSL Engine, the performance is similar regardless of QAT offloading, which indicates that OpenSSL will default to software as a fallback solution.

Workaround:

There is currently no workaround available. OpenSSL will still work, but without the performance increase from HW offloading.

Issue:

Pods requesting additional networks using SR-IOV CNI will fail to start

Detail:

The SR-IOV CNI needs access to the physical functions (PFs) on NICs attached to the VM Host. As these are not available in the VMs, SR-IOV CNI will fail to create pod interfaces and the pod will not be started.

Workaround:

There is currently no workaround available. VFs that are listed as allocatable resources can still be requested and added to pods, but the additional functionality of SR-IOV CNI such as IPAM and making the interface available in the pod will not work.

Issue:

Occasionally the sriov-network-device-plugin does not detect new or updated VF resources.

Detail:

There is a known issue with sriov-network-device-plugin where the service fails to detect new or updated VF resources if not available when the service creates its ConfigMap and loads the daemonset. See https://github.com/k8snetworkplumbingwg/sriov-network-device-plugin/issues/276.

Workaround:

Delete the sriov-device-plugin-pod and resources will be present when pod is automatically restarted.

Issue:

Collectd plugin fails to start.

Detail:

On some platforms, the collectd pod fails to start due to various plugin incompatibilities.

Workaround:
Disable problematic collectd plugins by adding to the exclude_collectd_plugins list in the Ansible host_vars configuration file.

Appendix B Abbreviations

The following abbreviations are used in this document.

ABBREVIATION	DESCRIPTION
AGF	Access Gateway Function
AIA	Accelerator Interfacing Architecture
AMX	Advance Matrix Multiply
BIOS	Basic Input/Output System
BMRA	Bare Metal Reference Architecture
BOM	Bill of Material
CA	Certificate Authority
CDN	Content Delivery Network
CLOS	Class of Service
CMTS	Cable Modem Termination System
CNF	Cloud Native Network Function
CNI	Container Network Interface
CO	Central Office
CRI	Container Runtime Interface
CXL	Compute Express Link
DDP	Dynamic Device Personalization
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name Service
DPDK	Data Plane Development Kit
DRAM	Dynamic Random Access Memory
DSA	Intel® Data Streaming Accelerator (Intel® DSA)
FP	Floating Point
FPGA	Field-Programmable Gate Array
FW	Firmware
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® 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)

ABBREVIATION	DESCRIPTION
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
ISA	Instruction Set Architecture
I/O	Input/Output
K8s	Kubernetes
KMS	Key Management Service (KMS)
LCC	Low Core Count
LLC	Last Level Cache
LOM	LAN on Motherboard
NFD	Node Feature Discovery
NFV	Network Function Virtualization
NIC	Network Interface Card
NTP	Network Time Protocol
NVM	Non-Volatile Memory
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
PMD	Poll Mode Driver
PXE	Preboot Execution Environment
QAT	Intel® QuickAssist Technology
QoS	Quality of Service
RA	Reference Architecture
RAS	Reliability, Availability, and Serviceability
RDT	Intel® Resource Director Technology
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
TAS	Telemetry Aware Scheduling
TDP	Thermal Design Power
TLS	Transport Layer Security
TME	Total Memory Encryption
TMUL	Tile Multiply
UEFI	Unified Extensible Firmware Interface
UPF	User Plane Function

ABBREVIATION	DESCRIPTION
vBNG	Virtual Broadband Network Gateway
vCMTS	Virtual Cable Modem Termination System
VF	Virtual Function
VMRA	Virtual Machine Reference Architecture
VNF	Virtual Network Function
VPP	Vector Packet Processing

0622/DN/WIT/PDF 721803-004US