

Quick Start Guide

Network and Cloud Edge Reference System Architectures on a Single Server



Develop and verify cloud native services on single server setup based on Intel's 4th Generation Intel® Xeon® Scalable Processor platform.

Introduction

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

This document is a quick start guide to configure the **Container Bare Metal Reference Architecture (BMRA)** or the **Virtual Machine Reference Architecture (VMRA)** on a single **4th Generation Intel® Xeon® Scalable Processor** based platform.

The RA has a variety of configuration profile settings for different network traffic workloads. **The quick start guide has selected the Remote Central Office-Forwarding Configuration Profile.** For details on this and other Configuration Profiles, refer to the User Guides listed in the [Reference Documentation](#) section.



Reference Documentation

The [Network and Cloud Edge Bare Metal Reference System Architecture User Guide](#) provides information and full set of installation instructions for a BMRA.

The [Network and Cloud Edge Virtual Machine Reference System Architecture User Guide](#) provides information and full set of installation instructions for a VMRA.

The [Network and Cloud Edge Reference System Architectures Portfolio User Manual](#) provides additional information for the Reference Architectures including a complete list of reference documents.

Other collaterals, including technical guides and solution briefs that explain in detail the technologies enabled in the Reference Architectures 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>.

Hardware and Software BOM

Following is the list of the hardware and software components that are required for setting up the system on a single server:

- Ansible host – laptop or server running a Unix base distribution
- 4th Generation Intel Xeon Scalable Processor based server – Quanta S6Q (S2EG3SEQ8B)
- Ethernet Controller Adapter – Intel® Ethernet Network Adapter E810-2CQDA2
- Software: <https://github.com/intel/container-experience-kits/>

For details of the software BOM for the **Remote Central Office-Forwarding Configuration Profile**, refer to the BMRA and VMRA User Guides listed in the [Reference Documentation](#) section.

The **Remote Central Office-Forwarding Configuration Profile** normally enables QAT. Currently the QAT drivers for the 4th Generation Intel Xeon Scalable Processor are under NDA, so the following instructions include disabling QAT.

DSA, DLB, SRIOV SW are also under NDA but are not enabled in the **Remote Central Office-Forwarding Configuration Profile**.

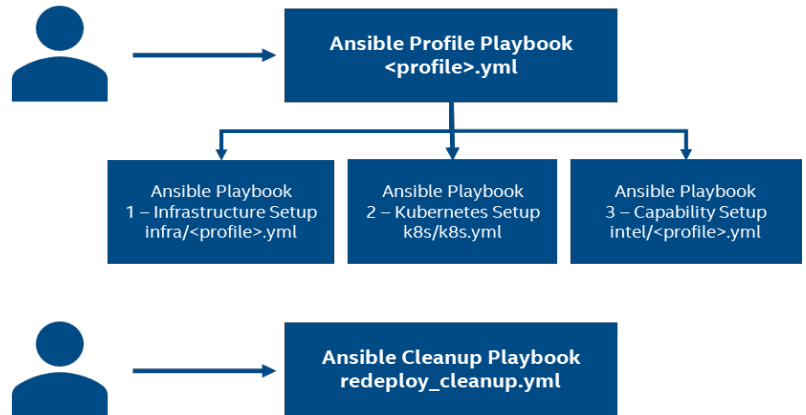
To enable the advance **4th Generation Intel Xeon Scalable Processor** features, please contact your Intel Representative.

Getting Started

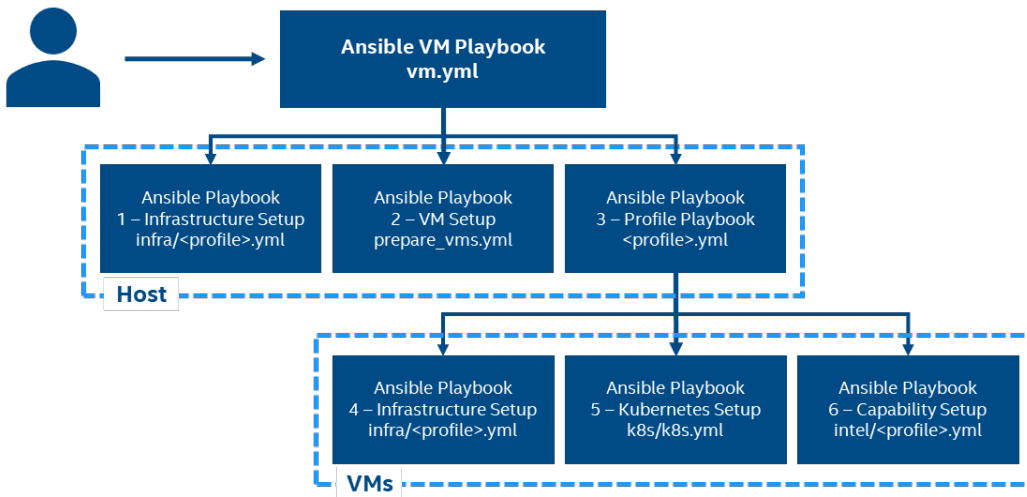
Ansible playbooks are used to install the Bare Metal (BMRA) and Virtual Machine (VMRA).

BMRA - provisioning is split into three steps:

1. **Infrastructure**
2. **Kubernetes**
3. **Capabilities**



VMRA - installation is split between the host and the VMs. On the host, the steps are: **Infrastructure** and **VM Setup**. Once VMs have been created, the Ansible inventory is updated, and the same three steps used for BMRA are repeated for the virtual machines (VMs).



Step 1 - Set Up the System

Refer to [Network and Cloud Edge Bare Metal Reference System Architecture User Guide](#) or [Network and Cloud Edge Virtual Machine Reference System Architecture User Guide](#): Section 6.1.

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

Ansible Host

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

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

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

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

3. Copy the public key to the target server:

```
# ssh-copy-id root@<target IP>
```

4. Verify passwordless connectivity to the target server:

```
# ssh root@<target IP>
```



Network and Cloud Edge Reference System Architecture on a Single Server – Quick Start Guide

Target Server

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

```
# sudo apt install -y python3 openssh-server lshw
```

2. As part of the configuration in Step 3, information about PCI devices for SR-IOV must be specified.

3. Find the relevant PCI IDs (bus:device.function) using 'lspci', and note down the IDs for later when configuring host_vars on the Ansible host:

```
# lspci | grep Eth
18:00.0 Ethernet controller: Intel Corporation Ethernet Controller E810-C for QSFP (rev 01)
18:00.1 Ethernet controller: Intel Corporation Ethernet Controller E810-C for QSFP (rev 01)
```

Target Server - Additional steps only needed for VMRA

For VMRA information about PCI, IDs of Virtual Functions (VFs) must be specified for the configuration of VMs.

1. Using device 18:00.0 as an example, create VFs for the device:

```
# echo "2" > /sys/bus/pci/devices/0000\:18\:00.0/sriov_numvfs
```

2. Find the relevant PCI IDs for the SR-IOV VFs:

```
# lspci | grep Eth | grep Virtual
18:01.0 Ethernet controller: Intel Corporation Ethernet Adaptive Virtual Function (rev 01)
18:01.1 Ethernet controller: Intel Corporation Ethernet Adaptive Virtual Function (rev 01)
```

Step 2 - Download and Install

Refer to [Network and Cloud Edge Bare Metal Reference System Architecture User Guide](#) or [Network and Cloud Edge Virtual Machine Reference System Architecture User Guide](#): Section 2.5.

Ansible Host

1. Download the source code from GitHub repository for the RA server:

```
# git clone https://github.com/intel/container-experience-kits/
# cd container-experience-kits
# git checkout v22.05
# git submodule update --init
```

2. Install the requirements needed by the deployment scripts:

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

Download & Install



Ansible
Playbook

Step 3 – Configure

Refer to [Network and Cloud Edge Bare Metal Reference System Architecture User Guide](#) or [Network and Cloud Edge Virtual Machine Reference System Architecture User Guide](#): Section 10.3.

The **Remote Central Office-Forwarding** configuration profile is used for this deployment. The BMRA and VMRA configuration steps are different.

Configure BMRA

Ansible Host

1. Generate the configuration files:

```
# make k8s-profile PROFILE=remote_fp ARCH=spr
```

2. Update the **inventory.ini** file to match a one server deployment. The values for **<target hostname>** and **<target IP>** must be updated to match the target system.

```
# cd container-experience-kits
# vim inventory.ini

[all]
<target hostname> ansible_host=<target IP> ip=<target IP> ansible_user=root
localhost ansible_connection=local ansible_python_interpreter=/usr/bin/python3

[vm_host]

[kube_control_plane]
<target hostname>

[etcd]
<target hostname>
```

Configure



Network and Cloud Edge Reference System Architecture on a Single Server – Quick Start Guide

```
[kube_node]
<target hostname>

[k8s_cluster:children]
kube_control_plane
kube_node

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

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

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

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

4. Update `host_vars/<target_hostname>.yml` with PCI device information specific to the target server:

```
## host_vars/<target_hostname>.yml ##
dataplane_interfaces:
  - bus_info: "18:00.0" # Use the SR-IOV PCI ID here
    pf_driver: ice
    ddp_profile: "ice_comms-1.3.31.0.pkg"
    default_vf_driver: "iavf"
    sriov_numvfs: 6
    sriov_vfs: # This is optional, but can be used to change
the driver of specific VFs. Any VF not specified here will use the "default_vf_driver"
specified above.
      vf_00: "vfio-pci"
      vf_05: "vfio-pci"
```

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

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

5. Disable QAT features in `host_vars/<target_hostname>.yml`:

```
## host_vars/<target_hostname>.yml ##
update_qat_drivers: false
openssl_install: false
```

6. Disable QAT features in `group_vars/all.yml`

```
## group_vars/all.yml ##
openssl_engine_enabled: false
```

For information about enabling QAT, please contact your Intel Representative.

7. To use the Key Management Reference Application (KMRA) which is enabled by default, an API key must be requested from <https://api.portal.trustedservices.intel.com/provisioning-certification> (click on "Subscribe").

Once the API key has been acquired, update `group_vars/all.yml` with the primary key:

```
kmra_enabled: true
kmra_pccs_api_key: "ffffffffffffffffffffffffffffffff" # Replace with API Key
```

Note: Without the API Key, the KMRA application will not work. It can be disabled by setting `kmra_enabled: false`

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

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

9. (optional) Verify that Ansible can connect to the target server, by running the below command and checking the output generated in the `all_system_facts.txt` file:

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

10. Apply a patch for the Kubespray submodule:

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

Network and Cloud Edge Reference System Architecture on a Single Server – Quick Start Guide

Configure VMRA

Ansible Host

1. Generate the configuration files:

```
# make vm-profile PROFILE=remote_fp ARCH=spr
```

2. Update the **inventory.ini** file to match a one server deployment. The values for `<target hostname>` and `<target IP>` must be updated to match the target system. Do not remove or modify the commented lines with "vm-ctrl-1" and "vm-work-1".

```
# vim inventory.ini

[all]
<target hostname> ansible_host=<target IP> ip=<target IP> ansible_user=root
localhost ansible_connection=local ansible_python_interpreter=/usr/bin/python3

[vm_host]
<target hostname>

[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
```

3. Update the filename of the `host_vars` file for the target machine:

```
# mv host_vars/host-for-vms-1.yml host_vars/<target hostname>.yml
```

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

Due to the additional layer of virtualization through the VMs, both `host_vars/<target hostname>.yml` and `host_vars/vm-work-1.yml` will need to be updated with PCI information:

```
## host_vars/<target hostname>.yml ##
dataplane_interfaces:
  - bus_info: "18:00.0"          # Use the SR-IOV PCI ID here
    pf_driver: ice
    ddp_profile: "ice_comms-1.3.31.0.pkg"
    default_vf_driver: "vfio-pci" # The driver here should always be "vfio-pci"
    sriov_numvfs: 6

vms:
  - type: "ctrl"
    (...)
  - type: "work"
    (...)
  pci:
    - "18:01.0"          # Use the SR-IOV VF PCI ID(s) here
    - "18:01.1"          # Use the SR-IOV VF PCI ID(s) here
```

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

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

5. After updating the PCI assignment to the worker VM, the configuration in `host_vars/vm-work-1.yml` must be updated to use these devices. Take note of the PCI IDs used here, as they follow the order in which the devices are assigned to the VM above. Using this example, the VF with ID "18:01.0" above is the first in the list, which will correspond to "06:00.0" inside of the VM as shown below. Similarly, "18:01.1" will be "07:00.0" in the VM.

```
## host_vars/vm-work-1.yml ##
dataplane_interfaces:
  - bus_info: "06:00.0"
    pf_driver: iavf          # Do not modify the value of "pf_driver"
    sriov_numvfs: 0         # Do not modify the value of "sriov_numvfs"
```

Network and Cloud Edge Reference System Architecture on a Single Server – Quick Start Guide

```
default_vf_driver: "igb_uio"          # Do not modify the value of "default_vf_driver"
- bus_info: "07:00.0"
  pf_driver: iavf
  sriov_numvfs: 0
default_vf_driver: "igb_uio"
```

6. Disable QAT features in `host_vars/<target_hostname>.yml` and `host_vars/vm-work-1.yml`:

```
## host_vars/<target_hostname>.yml & host_vars/vm-work-1.yml ##
update_qat_drivers: false
openssl_install: false
```

7. Disable QAT features in `group_vars/all.yml`

```
## group_vars/all.yml ##
openssl_engine_enabled: false
```

For information about enabling QAT, please contact your Intel Representative.

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

8. (optional) Verify that Ansible can connect to the target server, by running the below command and checking the output generated in the `all_system_facts.txt` file:

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

9. Apply a patch for the Kubespray submodule:

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

Step 4 – Deploy

Refer to [Network and Cloud Edge Bare Metal Reference System Architecture User Guide](#) or [Network and Cloud Edge Virtual Machine Reference System Architecture User Guide](#): Section 2.5.5.

Ansible Host

Now the RA can be deployed. The command used depends on the choice between [BMRA] and [VMRA] above:

```
[BMRA] # ansible-playbook -i inventory.ini playbooks/remote_fp.yml
[VMRA] # ansible-playbook -i inventory.ini playbooks/vm.yml
```

Deploy



Step 5 – Validate

Refer to [Network and Cloud Edge Bare Metal Reference System Architecture User Guide](#) or [Network and Cloud Edge Virtual Machine Reference System Architecture User Guide](#): Section 5.

Ansible Host

1. To interact with the Kubernetes CLI (kubectl), start by connecting to a controller node in the cluster, which can be done using the below commands for either BMRA or VMRA:

```
(BMRA) # ssh root@<target ip>
(VMRA) # ssh vm-ctrl-1          # Uses an alias specified in /root/.ssh/config
```

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

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

Validate



Additional feature verification tests can be found here:

<https://github.com/intel/container-experience-kits/tree/master/verification>

Document Revision History

REVISION	DATE	DESCRIPTION
001	May 2022	Initial release.
002	June 2022	Added more references in the Getting Started section.



No product or component can be absolutely secure.

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

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

© Intel Corporation. Intel, the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries. Other names and brands may be claimed as the property of others.