

SUSE Telco Cloud, Ampere Platforms

SUSE® Telco Cloud on Ampere® Platforms

Density, performance & TCO gains for telecom operators

SUSE Telco Cloud
Ampere Platforms

Alberto Morgante, Principal Telco Engineer (SUSE)
Kristian Zhelyazkov, Principal Software Engineer (SUSE)
Tim Irnich, Senior Distinguished Telco Product Manager (SUSE)
Fatih Degirmenci, Senior Manager Core Infrastructure QE (SUSE)
Sean Varley, Chief Evangelist and VP of Business Development (Ampere Computing)
Dave Neary, Director of Developer Relations (Ampere Computing)

SUSE® Telco Cloud on Ampere® Platforms

Density, performance & TCO gains for telecom operators

Date: 2026-02-17

Summary

SUSE® Telco Cloud on Ampere® Platforms provides a cloud-native, CPU-based infrastructure optimized for modern telecom workloads such as 5G, Open RAN, and edge computing. It delivers high performance, deterministic latency, and superior energy efficiency while eliminating reliance on specialized hardware accelerators. This solution reduces total cost of ownership, simplifies operations, and enables telecom operators to deploy scalable, future-ready networks

Disclaimer

Documents published as part of the series SUSE Technical Reference Documentation have been contributed voluntarily by SUSE employees and third parties. They are meant to serve as examples of how particular actions can be performed. They have been compiled with utmost attention to detail. However, this does not guarantee complete accuracy. SUSE cannot verify that actions described in these documents do what is claimed or whether actions described have unintended consequences. SUSE LLC, its affiliates, the authors, and the translators may not be held liable for possible errors or the consequences thereof.

Contents

- 1 Introduction 4
- 2 Structural transformation of the telecommunications landscape 5
- 3 Operational constraints and the Radio Access Network (RAN) 6
- 4 Architectural design for cloud-native telco 7
- 5 BIOS prerequisites 9
- 6 Edge Image Builder 12
- 7 Deployment details 14
- 8 Performance testing 19
- 9 Conclusion and next steps 24
- 10 Appendix 25
- 11 Legal notice 31
- 12 GNU Free Documentation License 32

1 Introduction

The telecommunications industry is rapidly evolving with the rollout of 5G-Advanced, Open RAN, Multi-access Edge Computing (MEC), and AI-driven automation — all of which are significantly increasing network traffic, processing demands, and performance expectations. To meet these challenges, telecom operators require cloud-native infrastructure that delivers high compute density, deterministic performance, and strong energy efficiency, especially in power- and space-constrained edge environments.

The collaboration between SUSE and Ampere Computing addresses these needs by combining optimized telco software with high-density, energy-efficient compute platforms. This approach enables scalable, vendor-neutral infrastructure that improves performance while reducing total cost of ownership.

1.1 Scope

This document defines the reference architecture and deployment framework for running cloud-native telco workloads with optimized performance, power efficiency, and scalability at the edge. It covers infrastructure configuration, system tuning, automated provisioning, and validation of the joint solution from SUSE and Ampere Computing.

1.2 Audience

This reference configuration is intended for:

- **telecommunications network architects, infrastructure and platform engineers, and operations teams** responsible for designing, deploying, and managing 5G and AI-enabled telco edge infrastructure.
- **technical decision-makers and IT or network leadership** evaluating platform modernization strategies to reduce infrastructure complexity, power consumption, and total cost of ownership.
- **system integrators, solution architects, and partner organizations** involved in designing, implementing, or validating telco edge solutions.

1.3 Acknowledgments

The authors wish to acknowledge the contributions of the following individuals to the creation of this document:

- Jivaji Ihare, Partner Solution Architect, SUSE
- Suresh S, Partner Solution Architect, SUSE
- Terry Smith, Ecosystem Solution Innovation Director, SUSE

2 Structural transformation of the telecommunications landscape

Telecommunications operators are undergoing the most profound transformation in the history of the industry. The simultaneous rollout of **5G-Advanced**, **Open RAN**, **massive Multi-access Edge Computing (MEC)**, and the **integration of AI-driven automation** are driving exponential growth in traffic, processing requirements, and service-level expectations. The demand for agility, scalability, and cost-efficiency is pushing the industry toward cloud-native software stacks running on high-performance, energy-efficient, general-purpose processors. The convergence of **high-performance hardware** and **optimized software stacks** is crucial for success in the modern telecommunications industry.

The strategic collaboration between SUSE and Ampere Computing addresses these challenges by validating **SUSE Telco Cloud** (<https://www.suse.com/products/edge-for-telco/>) [↗] configurations on **Ampere Platforms** (<https://amperecomputing.com/products/platforms>) [↗]. Our partnership represents a milestone in providing telecommunications operators with a **horizontally scalable, vendor-neutral infrastructure** that minimizes TCO while maximizing compute density in space-constrained edge environments. The move toward **disaggregated, multi-vendor ecosystems** requires a consistent underlying platform that can handle the **deterministic latency** and **high-throughput requirements** of modern 5G networks.

3 Operational constraints and the Radio Access Network (RAN)

Telecom operators face significant challenges when deploying RAN workloads, which require a combination of performance, latency, and power efficiency. The increasing demand for high-bandwidth 5G services has led to a **surge in RAN traffic**, making it essential to pack more compute resources into smaller footprints while optimizing power consumption. These workloads are no longer purely traditional packet-forwarding tasks. Modern Distributed Units (DU), Central Units (CU), User Plane Functions (UPF), and near-RT RIC platforms now embed **significant machine-learning inference for beam management, load prediction, slice assurance, and security analytics**. The net result is a dramatic increase in required FLOPS per site. Most current deployments still achieve this through power-hungry discrete accelerators (FPGAs, SmartNICs, and GPUs) that add cost, thermal complexity, and supply-chain fragility.

Energy efficiency has become a hard operational constraint. Many edge sites are limited on power and cooling, while electricity prices in many regions have risen **50-100% in the last three years**. Sustainability commitments further amplify the pressure where operators must materially reduce Scope 2 emissions while scaling capacity by orders of magnitude. Traditional high-TDP x86 platforms, even when paired with inline accelerators, routinely exceed 800-1200W per node under real-world vRAN loads — a trajectory that is no longer economically or environmentally viable at scale.

At the same time, the industry's move toward disaggregated, multi-vendor ecosystems exposes the limitations of closed, accelerator-dependent designs. Long integration cycles, firmware inter-dependencies, and single-source inline cards slow down lab validation and field upgrades, while creating hidden TCO penalties that often exceed the original hardware savings.

4 Architectural design for cloud-native telco

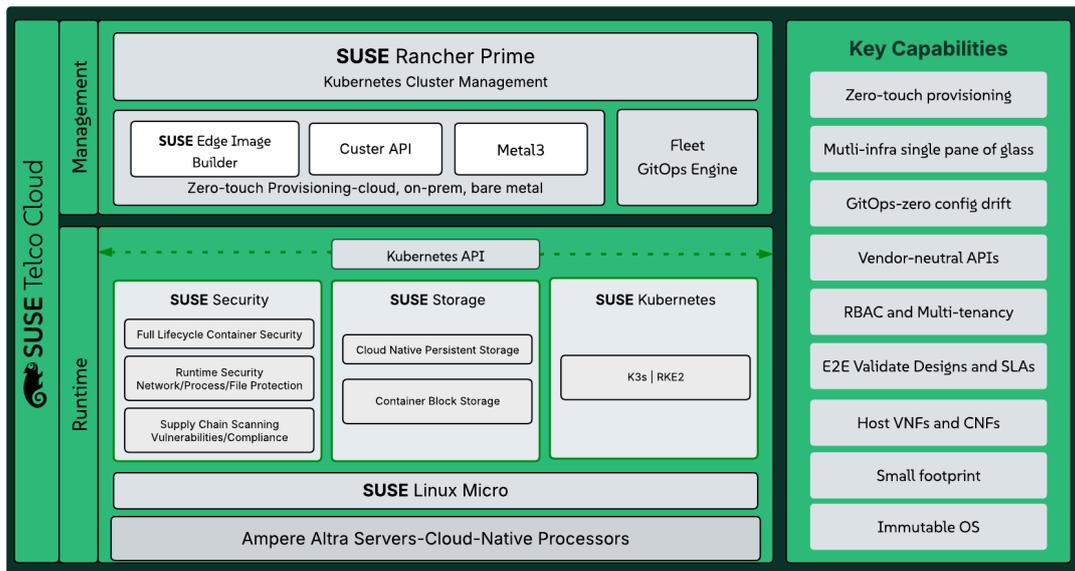


FIGURE 1: SUSE TELCO CLOUD ON AMPERE ALTRA – REFERENCE ARCHITECTURE

This joint reference architecture by SUSE and Ampere Computing addresses these challenges with a **radically simplified, high-efficiency, CPU-only platform** built on Arm-based architecture silicon.

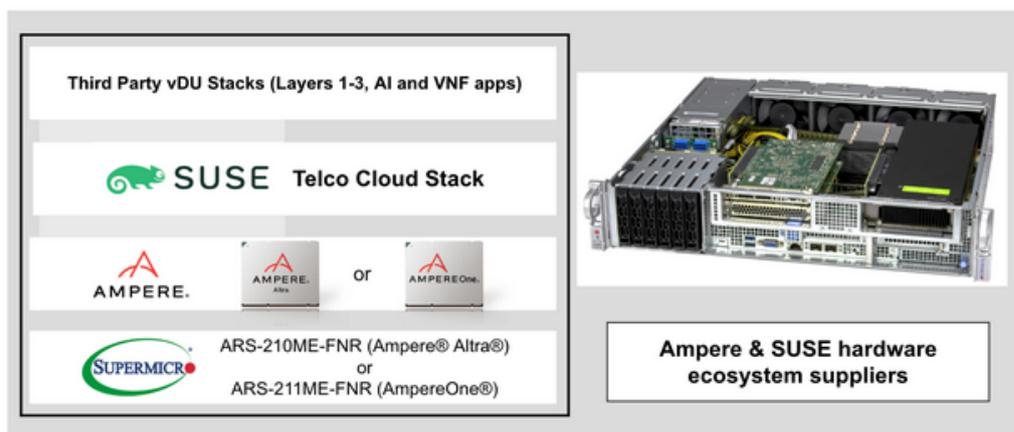


FIGURE 2: SUSE | AMPERE REFERENCE ARCHITECTURE FOR TELCO CLOUD AND RAN

Ampere processors are built for high performance while delivering excellent power efficiency. The processors scale linearly with load thereby enhancing power efficiency for workloads that do not require the full core capacity of these high core count products. This characteristic al-

so creates **natural headroom for new applications or AI algorithms** to expand functionality easily in power and space constrained environments like the telco RAN. Finally, Ampere processors are **highly predictable** yielding very deterministic performance for parallel scale-out workloads like packet processing, SW-only L1 processing, L2/L3 stacks and the sensitive AI functionality many carriers want to deploy in their networks.

Ampere processors deliver up to 192 arm64 cores in a single socket with a **completely uniform memory hierarchy** and **no performance variability across cores**. Combined with the SUSE near RT OS for telcos, this reference design addresses noisy-neighbor effects and provides the deterministic instruction retirement required for DPDK-based Layer 1 and Layer 2 processing in vRAN workloads.

Utilizing the Ampere Platforms's unparalleled advantages in core density and power efficiency, this solution is designed to **pack more compute resources into smaller footprints while minimizing power consumption**. This makes it an ideal fit for RAN deployments where space constraints are a significant challenge. The support of Arm-based hardware is a core pillar of SUSE's telco strategy because of the undeniable advantages in terms of core density and power efficiency.

The SUSE Telco Cloud solution is specifically designed to take advantage of these benefits by providing a **cloud-native infrastructure that is optimized for RAN workloads**. Our combined solution includes support for containerization, microservices orchestration, and network function virtualization (NFV) – all essential components for deploying modern 5G networks. By using SUSE Telco Cloud on the Ampere Altra platform, telecom operators can enjoy a significant reduction in total cost of ownership while achieving better performance density, latency, and power efficiency.

Because the entire solution is purely CPU-based and fully open, operators **avoid accelerator lock-in, reduce validation time from months to weeks**. They can also **run any certified CNF or VNF on the same physical node**, whether it is a FlexRAN-style DU, an open source L1, a 5G Core UPF, or an edge AI inference application. The result is a horizontally scalable platform that **simultaneously maximizes site capacity, minimizes power and cooling requirements, and future-proofs the infrastructure for 6G-era workloads** that will demand even greater compute density at the edge.



Note

This reference design uses the Ampere® Platforms processor, featuring the aarch64x architecture with 128 cores. Ampere Altra systems tested here are widely used. This reference architecture is also relevant for AmpereOne® Platform systems, which provide even higher performance and core density.

5 BIOS prerequisites

5.1 Enable SR-IOV for the network interfaces

SR-IOV (Single Root I/O Virtualization) is a specification that allows a single Peripheral Component Interconnect Express (PCIe) physical device to present itself as multiple separate physical devices to a hypervisor (the "Single Root"). This is achieved by dividing the physical network interface card (NIC) into multiple virtual functions (VFs).

- **Physical Function (PF):** The full-featured PCIe function that includes the SR-IOV capability. It is typically owned by the hypervisor/host OS and used to configure and manage the VFs.
- **Virtual Function (VF):** A lightweight PCIe function that contains the resources necessary for data movement but has a restricted configuration space. VFs can be directly assigned to virtual machines (VMs) or containers, bypassing the host OS's network stack and the hypervisor's virtual switch.

SR-IOV is critically important for modern Telco Cloud and 5G/vRAN (Virtualized Radio Access Network) deployments because of its direct impact on network performance and deterministic latency:

- **Low Latency and High Throughput:** By allowing VFs to be directly assigned to Network Functions (like virtual Distributed Units - vDUs), SR-IOV creates a near bare-metal data path. This eliminates the overhead of the hypervisor's network stack and virtual switching layers, which significantly reduces latency and maximizes throughput—essential for handling high-bandwidth, real-time 5G traffic.
- **Deterministic Performance:** In 5G vRAN, L1 (Layer 1) processing requires very strict timing and deterministic performance. The bypass of the virtual switch layer minimizes "noisy neighbor" effects and jitter caused by context switching and software processing, ensuring a consistent and predictable packet arrival time crucial for maintaining quality of service.
- **Efficiency and CPU Offload:** SR-IOV offloads packet processing from the host CPU to the NIC hardware. This frees up valuable CPU cores—especially the real-time isolated cores (as configured in the document)—to focus solely on running the demanding application-level workloads (like DPDK processing or specialized AI inference) It prevents them from spending cycles on packet forwarding and network virtualization overhead.
- **Enabling Cloud-Native Functions (CNFs):** For containerized network functions (CNFs) running on Kubernetes (like SUSE Telco Cloud), SR-IOV is leveraged via mechanisms like the SR-IOV Network Operator to provide direct network connectivity to pods. This allows CNFs to meet the rigorous performance demands previously only achievable with specialized hardware.

The first task to enable SR-IOV is to configure it in the system BIOS (See [Supermicro FAQ: How to enable SR-IOV \(https://www.supermicro.com/support/faqs/faq.php?faq=38914\)](https://www.supermicro.com/support/faqs/faq.php?faq=38914)).

1. Enable SR-IOV for the specific NIC under *Advance > Chipset > North Bridge > IIO Config > Intel VT*.
2. Enable SR-IOV globally under *Advance > PCIe Config > SR-IOV*.

After configuring properly the BIOS, the following command can be executed to verify if the SR-IOV is enabled on the interfaces that we want to use:

```
find /sys/class/net/*/device/sriov_totalvfs -exec sh -c 'echo -n "{}':
```

```
"; cat \{\}' \;
```

This results in output similar to:

```
/sys/class/net/enP1pls0f0np0/device/sriov_totalvfs: 64  
  
/sys/class/net/enP1pls0f1np1/device/sriov_totalvfs: 64  
  
/sys/class/net/enP1pls0f2np2/device/sriov_totalvfs: 64  
  
/sys/class/net/enP1pls0f3np3/device/sriov_totalvfs: 64  
  
/sys/class/net/en01np0/device/sriov_totalvfs: 2  
  
/sys/class/net/en02np1/device/sriov_totalvfs: 2
```

5.2 BIOS tuning

After enabling SR-IOV on the desired NICs, the next step involves tuning the BIOS settings. This can be accomplished through the BMC Web UI with the following settings:

1. *Advanced > ACPI Settings > Enable CPPC* - Set to **Disabled**.
2. *Advanced > ACPI Settings > Enable LPI* - Set to **Disabled**.
3. *Chipset > CPU Configuration > ANC mode* Set to **Quadrant**.

Collaborative Processor Performance Control (CPPC) is an ACPI interface that allows the OS to manage CPU performance states more efficiently than legacy methods (like P-States and SpeedStep). Instead of using discrete frequency steps, CPPC communicates a continuous performance scale, allowing the CPU to change frequency/voltage faster and more precisely.

Low Power Idle (LPI) exposes the Low Power S0 Idle (also known as S0ix or Modern Standby) capability to the OS. It allows the CPU and specific components to enter ultra-low power states without fully sleeping (S3), maintaining a quick wake-up capability.

Ampere NUMA Control (ANC) is specific to Ampere Altra / Altra Max ARM processors and determines how the processor's cores and memory controllers are clustered into Non-Uniform Memory Access (NUMA) domains.

Quadrant implies 4 NUMA domains per socket, yielding lowest local memory latency.

6 Edge Image Builder

6.1 What is Edge Image Builder

The need for easily auditable, reproducible, and customized base images becomes paramount to the long-term stability and maintainability of the environment. [Edge Image Builder \(EIB\)](https://documentation.suse.com/suse-edge/3.3/html/edge/components-eib.html) (<https://documentation.suse.com/suse-edge/3.3/html/edge/components-eib.html>), one of the components of SUSE Telco Cloud, helps mitigate these concerns. It provides a mechanism to customize SL Micro base images to include all of the configurations and workload artifacts needed by telco deployments. [EIB is an open source project](https://github.com/suse-edge/edge-image-builder) (<https://github.com/suse-edge/edge-image-builder>) that enables the simplified and quick configuration of customized SUSE Linux Micro images for a variety of scenarios. The resulting image contains all of the configuration and binary artifacts (RPMs, container images, etc.) necessary to provision the node at first boot. This way, it provides true zero-touch provisioning even in the case of low-bandwidth connections and air-gapped scenarios. A single image can be reused across multiple nodes, including advanced situations such as deploying HA Kubernetes clusters and individualized network configuration per node. By leveraging a declarative, YAML-based definition format, EIB easily fits into existing GitOps and CI/CD pipelines, generating reproducible images with full accountability of their contents.

6.2 EIB configuration

At [SUSE Edge: telco-cloud-examples](https://github.com/suse-edge/telco-cloud-examples/tree/main/telco-examples/edge-clusters/dhcp-less/eib) (<https://github.com/suse-edge/telco-cloud-examples/tree/main/telco-examples/edge-clusters/dhcp-less/eib>) you can find some examples prepared for use in your SUSE Telco Cloud deployment. Below are further details on the configurations you need to properly deploy a downstream cluster for running telco workloads on the Ampere platform.

6.2.1 Image Definition File

The Image Definition File is a YAML document describing a single image to build. See the example below.

```
apiVersion: 1.0
image:
  imageType: RAW
  arch: x86_64
```

```
baseImage: SL-Micro.x86_64.raw
outputImageName: eibimage-slmicro-rt-telco.raw
operatingSystem:
  kernelArgs:
    - ignition.platform.id=openstack
  systemd:
    disable:
      - rebootmgr
      - transactional-update.timer
      - transactional-update-cleanup.timer
      - fstrim
      - time-sync.target
  users:
    - username: root
      encryptedPassword: ${ROOT_PASSWORD}
  packages:
    packageList:
      - jq
      - dpdk
      - dpdk-tools
      - libdpdk-23
      - pf-bb-config
      - open-iscsi
      - tuned
      - cpupower
      - openssh-server-config-rootlogin
  sccRegistrationCode: ${SCC_REGISTRATION_CODE}
```

The file is specified using the `--definition-file` argument. Only a single image may be built at a time. However, the same image configuration directory may be used to build multiple images by creating multiple definition files.

You can find more information at [Edge Image Builder: Building images \(https://github.com/suse-edge/edge-image-builder/blob/main/docs/building-images.md\)](https://github.com/suse-edge/edge-image-builder/blob/main/docs/building-images.md).

6.2.2 Custom files

EIB can bundle custom scripts that run during the combustion phase when a node is booted with the built image. However, custom files are not automatically deployed on the node when it boots. If custom files are needed beyond the combustion phase, a script should be included that explicitly copies them to the file system.

See [Section 10.1, "Sample custom performance tuning file"](#) for reference.

Combustion scripts are executed alphabetically. All scripts automatically included by EIB are prefixed using values between 00 and 49 (for example, `05-configure-network.sh`, `30-suma-register.sh`). Unless certain that you need to interrupt the default flow, all custom scripts should be prefixed within the range 50 to 99 (for example, `60-my-script.sh`) or not begin with a number.

See [Section 10.2, “Sample custom file to copy scripts to the file system”](#) for reference.

You can find more information at [Edge Image Builder: Building images \(https://github.com/suse-edge/edge-image-builder/blob/main/docs/building-images.md\)](https://github.com/suse-edge/edge-image-builder/blob/main/docs/building-images.md).

7 Deployment details

7.1 Directed network provisioning

Directed network provisioning is a feature that allows you to automate the provisioning of downstream clusters. This feature is useful when you have many downstream clusters to provision, and you want to automate the process.

The network architecture is based on the following components:

- **Management network**
This network is used for out-of-band management of downstream cluster nodes. This network is usually connected to a separate management switch, but it can be connected to the same service switch using VLANs to isolate the traffic.
- **Control-plane network**
This network is used for the communication between the downstream cluster nodes and the services that are running on them. It is also used for communication between the nodes and external services, like the DHCP and DNS. For connected environments, the switch/router can handle traffic through the Internet.
- **Other networks**
In some cases, nodes could be connected to other networks for specific purposes.

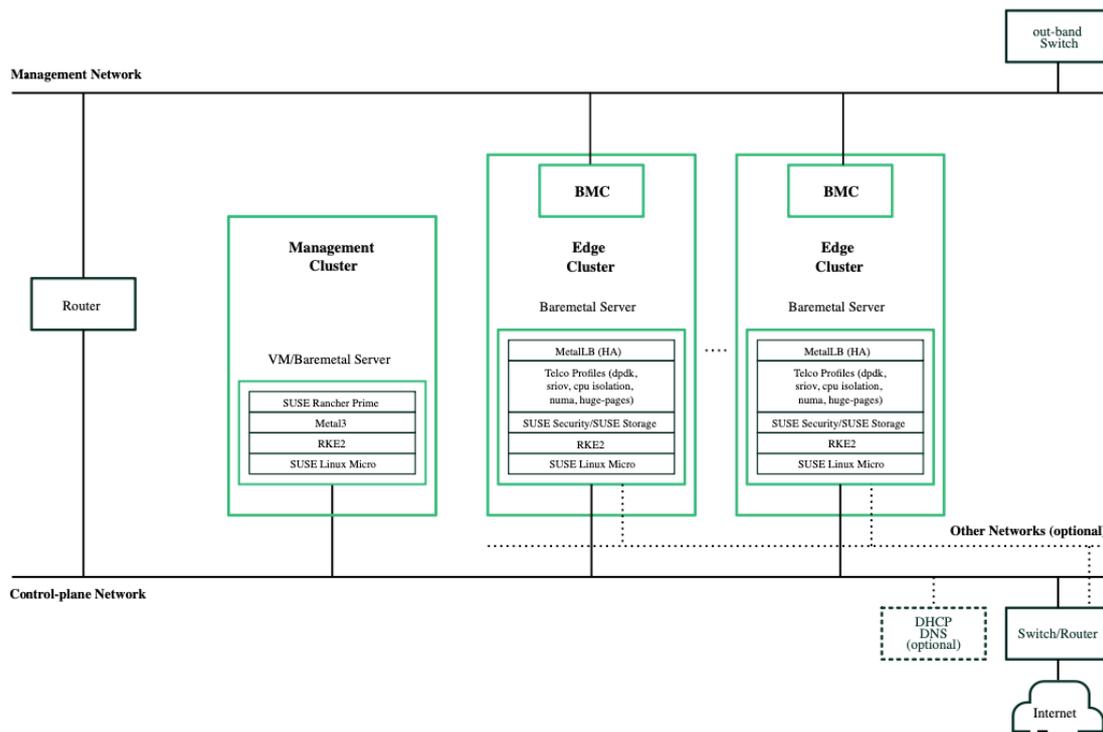


FIGURE 3: MANAGEMENT AND CONTROL-PLANE NETWORK ARCHITECTURE FOR TELCO CLOUD



Note

For this document, it is assumed that the management cluster is already deployed in order to focus on the downstream cluster for running telco performance tests.

7.1.1 Cluster API and SUSE Telco Cloud

Cluster API is a Kubernetes sub-project focused on providing declarative APIs and tooling to simplify provisioning, upgrading, and operating multiple Kubernetes clusters.

Started by the Kubernetes Special Interest Group (SIG), Cluster Lifecycle, the Cluster API project uses Kubernetes-style APIs and patterns to automate cluster lifecycle management for platform operators. The supporting infrastructure, like virtual machines, networks, load balancers, and VPCs, and the Kubernetes cluster configuration are all defined in the same way that application developers operate deploying and managing their workloads. This enables consistent and repeatable cluster deployments across a wide variety of infrastructure environments.

The following diagram illustrates the workflow used to deploy a single-node downstream cluster with directed network provisioning:

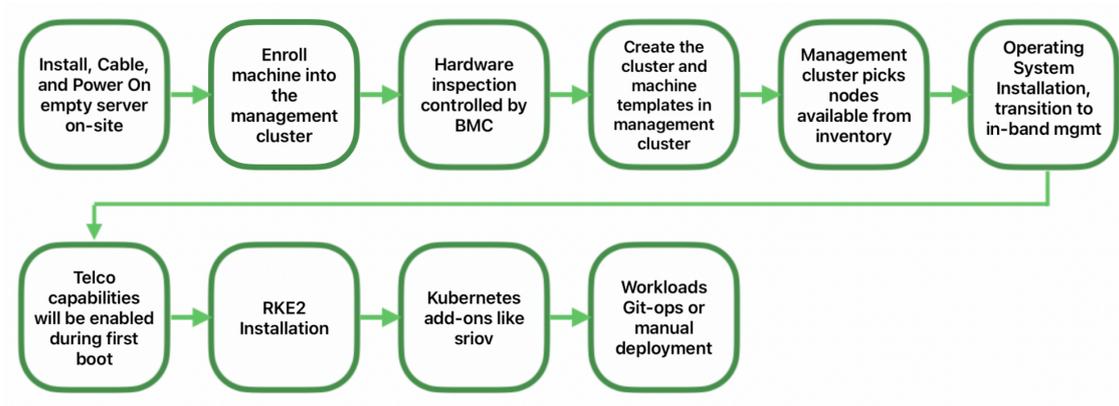


FIGURE 4: SINGLE-NODE DOWNSTREAM CLUSTER PROVISIONING WORKFLOW



Tip

The EIB-generated image described in the previous section must be located in the management cluster in order to set up the downstream cluster.

7.1.2 CAPI and Metal3 manifests

A full example manifest file, `telco-capi-single-node-sriov-auto.yaml`, for a single-node cluster is available in the [SUSE Edge: telco-cloud-examples](https://github.com/suse-edge/telco-cloud-examples/blob/main/telco-examples/edge-clusters/dhcp-less/edge-telco-single-node/telco-capi-single-node-sriov-auto.yaml) (<https://github.com/suse-edge/telco-cloud-examples/blob/main/telco-examples/edge-clusters/dhcp-less/edge-telco-single-node/telco-capi-single-node-sriov-auto.yaml>). The important sections of the CAPI and Metal3 manifests are outlined below.

- `Pre_rke2_commands`

This is a requirement to enable the possibility of creating VF (virtual functions) using SR-IOV

```
modprobe vfio-pci enable_sriov=1 disable_idle_d3=1
```

- `Kernel_arguments`

The kernel arguments shown below enable the real-time kernel to work properly and give the best performance and lowest latency for telco workloads. These options are described in detail in the [SUSE Telco Cloud Kernel arguments for low latency and high performance \(https://documentation.suse.com/suse-edge/3.5/html/edge/atip-features.html#kernel-args\)](https://documentation.suse.com/suse-edge/3.5/html/edge/atip-features.html#kernel-args).

```
- randomize_kstack_offset=0
  - rd.timeout=60
  - rd.retry=45
  - console=ttyS1,115200
  - console=tty0
  - hugepagesz=1G hugepages=20
  - hugepagesz=2M hugepages=0
  - default_hugepagesz=1G
  - ignition.platform.id=openstack
  - intel_iommu=on
  - iommu=pt
  - irqaffinity=0-5
  - isolcpus=domain,nohz,managed_irq,6-127
  - mce=off
  - nohz=on
  - net.ifnames=1
  - nmi_watchdog=0
  - nohz_full=6-127
  - nosoftlockup
  - nowatchdog
  - quiet
  - rcu_nocb_poll
  - rcu_nocbs=6-127
  - rcupdate.rcu_cpu_stall_suppress=1
  - rcupdate.rcu_expedited=1
  - rcupdate.rcu_normal_after_boot=1
  - rcupdate.rcu_task_stall_timeout=0
  - rcutree.kthread_prio=99
  - security=selinux
  - selinux=1
```

- **systemd units**

Fine tuning of the system after it has been deployed simplified with systemd unit files. This includes CPU partitioning, various performance settings, SR-IOV parameters. An example is provided below:

```
- name: cpu-partitioning.service
  enabled: true
  contents: |
```

```

[Unit]
Description=cpu-partitioning
Wants=network-online.target
After=network.target network-online.target
[Service]
Type=oneshot
User=root
ExecStart=/bin/sh -c "echo isolated_cores=6-127 > /etc/tuned/cpu-
partitioning-variables.conf"
ExecStartPost=/bin/sh -c "tuned-adm profile cpu-partitioning"
ExecStartPost=/bin/sh -c "systemctl enable tuned.service"
[Install]
WantedBy=multi-user.target
- name: performance-settings.service
enabled: true
contents: |
[Unit]
Description=performance-settings
Wants=network-online.target
After=network.target network-online.target cpu-partitioning.service
[Service]
Type=oneshot
User=root
ExecStart=/bin/sh -c "/opt/performance-settings/performance-settings.sh"
[Install]
WantedBy=multi-user.target
- name: sriov-custom-auto-vfs.service
enabled: true
contents: |
[Unit]
Description=SRIOV Custom Auto VF Creation
Wants=network-online.target rke2-server.target
After=network.target network-online.target rke2-server.target
[Service]
User=root
Type=forking
TimeoutStartSec=5400
ExecStart=/bin/sh -c '/bin/sh -c "while ! /var/lib/rancher/rke2/bin/kubectl
--kubeconfig=/etc/rancher/rke2/rke2.yaml wait --for condition=ready nodes --
timeout=30m --selector=\'!node-role.kubernetes.io/control-plane\' ; do sleep
10 ; done"'
ExecStartPost=/bin/sh -c "/opt/sriov/sriov-auto-filler.sh"
RemainAfterExit=yes
KillMode=process
[Install]
WantedBy=multi-user.target

```

With the `telco-capi-single-node-sriov-auto.yaml` manifests file ready, execute the following command on the management cluster to start provisioning the new, downstream cluster with telco features:

```
telco-capi-single-node-sriov-auto.yaml
```

8 Performance testing

This section is divided into two distinct phases:

1. Network Stress Analysis
2. Deterministic CPU Validation

The objective of the networking test is to characterize the limits of the System Under Test (SUT) network stack. It is not merely checking for connectivity, but is stressing the interface to identify its saturation point and stability under load. This includes:

- **Bandwidth saturation**
Use iPerf3 to generate synthetic TCP traffic that exceeds the theoretical limit of the link. This forces the SUT to manage buffer flows and flow control, revealing the maximum sustainable throughput.
- **Stability analysis**
Inject UDP traffic streams to measure jitter (variance in packet arrival time) and packet loss. This simulates real-time data streams, like video or sensor data, where consistency is more critical than raw speed.
- **Latency profiling**
Run concurrent ICMP (ping) probes during load testing to measure the round-trip time (RTT). This determines how quickly the network stack can process simple requests while under heavy stress.

The objective of the CPU test is not to measure how fast the processor computes data (benchmarking), but rather how predictably it responds to events (determinism). This includes:

- **Latency Histogramming**

Use `cyclictest` to generate a high-frequency loop that sleeps and wakes up at a specific interval. Calculate the latency by comparing the scheduled wake-up time with the actual wake-up time. Run this for an extended period to catch rare "spikes" caused by hardware interrupts or kernel locks.

- Load simulation

Use `rt-app` to simulate a synthetic real-time application payload. This ensures the scheduler is being tested under conditions that mimic a production environment.

- Threshold validation

Analyze the maximum latency outlier. The strict success criterion is that no single event may be delayed by more than 10 μ s. Any spike above this threshold indicates a failure in the real-time configuration (such as, non-preemptible kernel sections or power management interference).

The following sections outline the operational strategy for each phase.

8.1 Network performance

These automated tests utilize `iperf3` to send traffic at a rate of 250 Mbps with a 64-byte packet size.

Our testing methodology involves creating two Pods: one configured as a server and the other as a client. We then initiate traffic from the client to the server and subsequently measure packet loss. This environment allows us to test four distinct scenarios:

1. Two nodes without SR-IOV
2. Single node without SR-IOV
3. Single node with SR-IOV
4. Two nodes with SR-IOV



FIGURE 5: NETWORK PERFORMANCE TEST SCENARIOS WITH AND WITHOUT SR-IOV

 Note

For the test to be considered successful, the packet drop rate must be below 1%.

The following is example output from one of the scenarios:

```
Connecting to host 192.168.5.137, port 5201
[ 5] local 192.168.19.217 port 33947 connected to 192.168.5.137 port 5201
[ ID] Interval          Transfer      Bitrate      Total Datagrams
[ 5]  0.00-1.00    sec  8.69 MBytes  72.8 Mbits/sec  142395
[ 5]  1.00-2.00    sec  8.80 MBytes  73.8 Mbits/sec  144138
[ 5]  2.00-3.00    sec  8.78 MBytes  73.7 Mbits/sec  143865
[ 5]  3.00-4.00    sec  8.78 MBytes  73.6 Mbits/sec  143841
[ 5]  4.00-5.00    sec  8.77 MBytes  73.6 Mbits/sec  143732
[ 5]  5.00-6.00    sec  8.78 MBytes  73.6 Mbits/sec  143851
[ 5]  6.00-7.00    sec  8.78 MBytes  73.6 Mbits/sec  143822
[ 5]  7.00-8.00    sec  8.77 MBytes  73.6 Mbits/sec  143752
[ 5]  8.00-9.00    sec  8.78 MBytes  73.6 Mbits/sec  143821
[ 5]  9.00-10.00   sec  8.79 MBytes  73.7 Mbits/sec  144057
-----
[ ID] Interval          Transfer      Bitrate      Jitter      Lost/Total Datagrams
[ 5]  0.00-10.00   sec  87.7 MBytes  73.6 Mbits/sec  0.000 ms    0/1437274 (0%) sender
[ 5]  0.00-10.00   sec  87.7 MBytes  73.6 Mbits/sec  0.011 ms   162/1437274 (0.011%) receiver

iperf Done.
```

8.2 CPU performance

The following tools are used for CPU performance testing:

- [rt-app \(https://github.com/scheduler-tools/rt-app\)](https://github.com/scheduler-tools/rt-app)
- [cyclictst \(https://documentation.suse.com/sle-rt/15-SP7/html/SLE-RT-all/article-hardware-testing.html\)](https://documentation.suse.com/sle-rt/15-SP7/html/SLE-RT-all/article-hardware-testing.html)

The test involves generating a 95% load using `rt-app`, then executing two, distinct testing scenarios, each lasting two hours:

1. Isolated Core Capacity

Measures the system's ability to schedule new tasks on the isolated cores when they are under a heavy load (95% utilization using `rt-app`).

2. System Under Stress

Measures the performance of the isolated cores while the housekeeping cores are experiencing a high load (95% utilization using `rt-app`). This simulates potential system issues.



Note

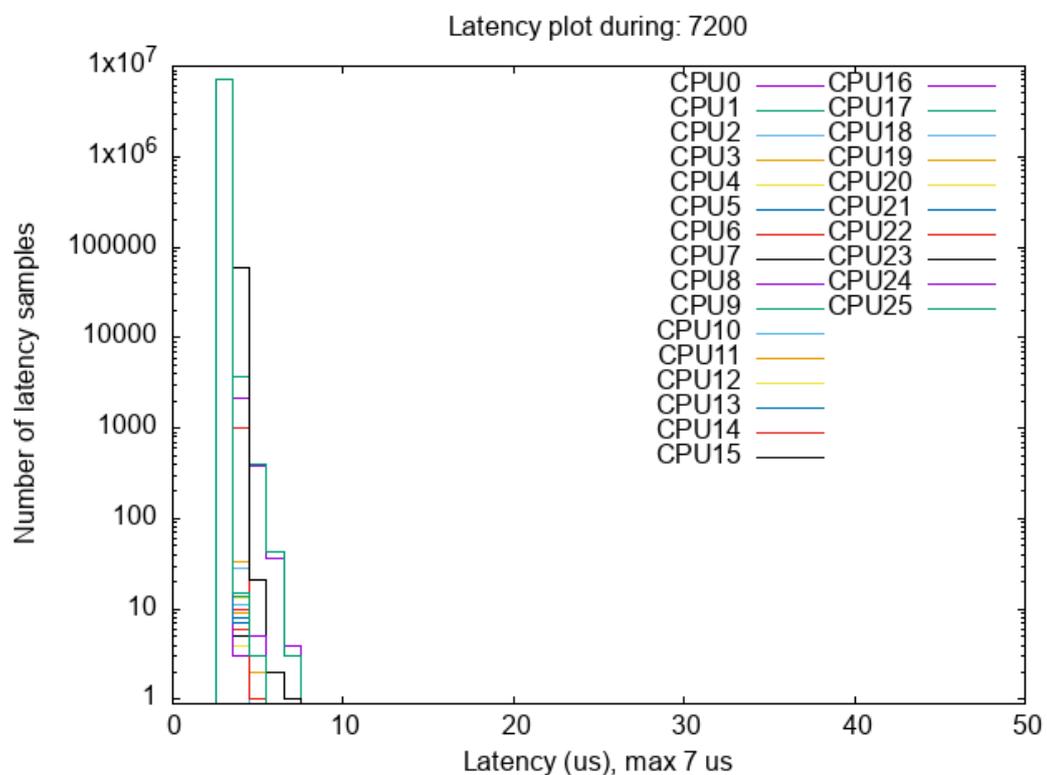
For the testing to be considered successful, the maximum latency of each core should be less than 10 us.

The tests are launched with the following command:

```
cyclictest -F /tmp/cyclictest-monitor.pipe -p98 -t36
--affinity=<cpu-isolated-cores> --mainaffinity=1 :--interval=1000
--histogram=60 --spike=10 --duration=7200 --quiet
```

Output from one of the tests is provided below.

- CPU latency performance under system stress



- Maximum latencies

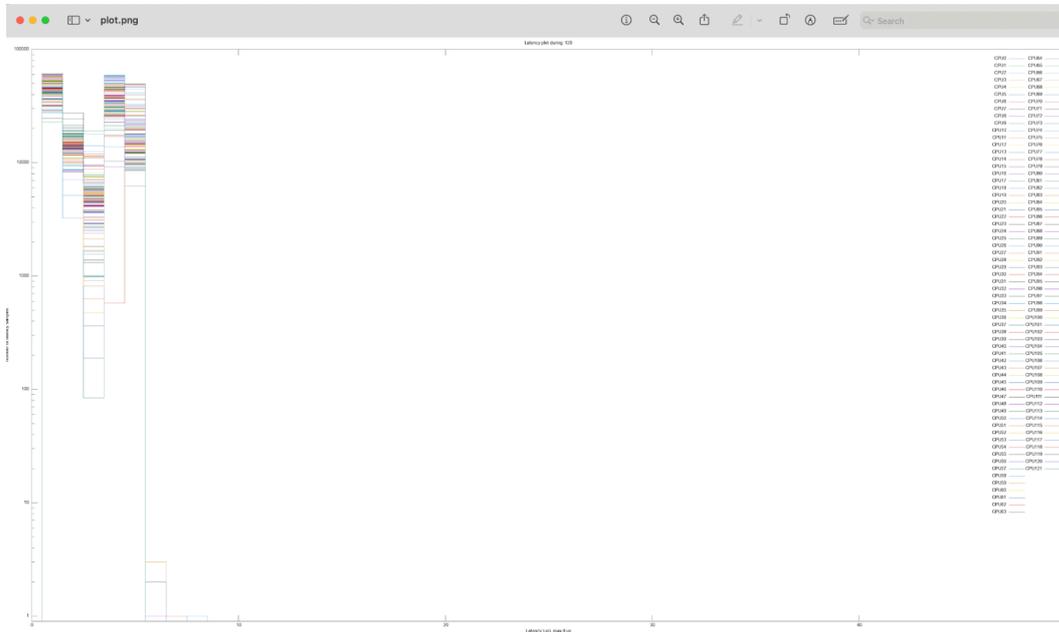
```
# Max Latencies: 00006 00006 00006 00006 00007 00006 00006 00006 00007 00006
00006 00006 00007 00005 00006 00006 00006 00006 00006 00007 00009 00008 00008
00008 00005 00006 00006 00007 00006 00006 00006 00006 00006 00006 00006 00007
00006 00006 00007 00006 00006 00006 00006 00006 00006 00008 00008 00008 00008
00008 00005 00006 00006 00005 00006 00006 00006 00005 00006 00009 00009 00008
00008 00008 00008 00008 00008 00006 00006 00006 00006 00006 00006 00006 00006
```

```

00008 00006 00006 00006 00007 00008 00006 00006 00005 00008 00008 00008 00008
00006 00007 00007 00006 00006 00006 00006 00006 00008 00008 00008 00005 00006
00006 00006 00006 00006 00006 00006 00007 00007 00008 00007 00007 00007 00007
00007 00006 00006 00006 00006 00006 00006 00006

```

- CPU latency distribution



9 Conclusion and next steps

The Ampere and SUSE joint solution delivers a future-proof, open, cloud-native telco platform. It combines unmatched core density, best-in-class power efficiency, and carrier-grade real-time performance, all without the need for additional accelerators or vendor lock-in.

Operators can now deploy high-capacity 5G RAN and edge services at dramatically lower power, space, and total cost while preserving full flexibility to run any CNF or VNF on the same infrastructure.

For your next steps:

- Explore the [SUSE Telco Cloud documentation \(https://documentation.suse.com/suse-edge/3.5/html/edge/\)](https://documentation.suse.com/suse-edge/3.5/html/edge/).
- Schedule a joint proof-of-concept or sizing workshop with your [SUSE \(https://www.suse.com/contact/\)](https://www.suse.com/contact/) and [Ampere Computing \(https://amperecomputing.com/company/contact-us/\)](https://amperecomputing.com/company/contact-us/) account teams.

10 Appendix

10.1 Sample custom performance tuning file

```
- name: performance-settings.sh
  content: |
    #!/bin/bash

    # Script generate to tuning the performance of the system for running Telco
    Workloads
    # This script is intended to be run on a worker node in a Telco Edge Cluster

    if [ "`whoami`" != "root" ]; then
        echo root required to run the script
        exit 127
    fi

    MAX_EXIT_LATENCY=1
    input=$(cat /etc/tuned/cpu-partitioning-variables.conf)
    total_cores=$(grep -c ^processor /proc/cpuinfo)
    cpus=$(echo "$input" | awk -F=' ' '{print $2}')

    expand_ranges() {
        echo "$1" | awk -v RS=',' ' '
        {
            if ($1 ~ /-/) {
                split($1, range, "-")
                for (i = range[1]; i <= range[2]; i++) {
                    printf i (i==range[2] ? "" : " ")
                }
            } else {
                printf $1
            }
            printf (NR == NR ? " " : "")
        }' | sed 's/ $//'
    }

    ISOLATED_CPUS=$(expand_ranges "$cpus")
    all_cores=$(seq 0 $((total_cores-1)))
    isolated_set=$(echo $ISOLATED_CPUS | tr ',' '\n') # Convert to newline separated
list
    HK_CPUS=""
```

```

for core in $all_cores; do
    if ! echo "$isolated_set" | grep -q "^$core$"; then
        HK_CPUS+=" $core "
    fi
done

HK_CPUS=$(echo "$HK_CPUS" | sed 's/ $//')

set_cpufreq_performance() {
    echo Configure: CpuFreq performance
    cpupower frequency-set -g performance | grep -v "Setting cpu:"
    cpupower set -b 0
}

unset_timer_migration() {
    echo Configure: Disable Timer migration
    sysctl kernel.timer_migration=0
}

migrate_kdaemons_hk() {
    echo Configure: Migrate kthreads to HK
    for NODE in `ls -l -d /sys/devices/system/node/node* | sed -e 's/.*node//'; do
        for KTHREAD in kswapd$NODE kcompactd$NODE ; do
            PID_KTHREAD=`pidof $KTHREAD`
            [ "$PID_KTHREAD" = "" ] && PID_KTHREAD=`pidof -w $KTHREAD`
            if [ "$PID_KTHREAD" = "" ]; then
                echo "WARNING: Unable to identify PID of $KTHREAD"
                continue
            fi
            taskset -pc `echo $HK_CPUS | tr ' ' ','` $PID_KTHREAD
        done
    done
}

set_isolatecpu_latency() {
    echo Configure: IsolCpus latency requirements
    cat /proc/cmdline | tr ' ' '\n' | grep -q ^idle=poll
    if [ $? -eq 0 ]; then
        echo "WARNING: Using idle=poll as a kernel paramter makes per-cpu pm qos
redundant"
        return
    fi
}

for CPU in $ISOLATED_CPUS; do
    SYSFS_PARAM="/sys/devices/system/cpu/cpu$CPU/power/pm_qos_resume_latency_us"
    if [ ! -e $SYSFS_PARAM ]; then
        echo "WARNING: Unable to set PM QOS max latency for CPU $CPU\n"
    fi
done

```

```

        continue
    fi
    echo $MAX_EXIT_LATENCY > $SYSFS_PARAM
    echo "Set PM QOS maximum resume latency on CPU $CPU to ${MAX_EXIT_LATENCY}us"
done
}

delay_vmstat_updates() {
    echo Configure: Delay vmstat updates
    sysctl -w vm.stat_interval=300
}

set_cpufreq_performance
unset_timer_migration
migrate_kdaemons_hk
set_isolatecpu_latency
delay_vmstat_updates

- name: sriov-auto-filler.sh
content: |
#!/bin/bash
cat <<- EOF > /var/sriov-networkpolicy-template.yaml
apiVersion: sriovnetwork.openshift.io/v1
kind: SriovNetworkNodePolicy
metadata:
  name: atip-RESOURCE_NAME
  namespace: sriov-network-operator
spec:
  nodeSelector:
    feature.node.kubernetes.io/network-sriov.capable: "true"
  resourceName: RESOURCE_NAME
  deviceType: DRIVER
  numVfs: NUMVF
  mtu: 1500
  nicSelector:
    pfNames: ["PFNAMES"]
    deviceID: "DEVICEID"
    vendor: "VENDOR"
    rootDevices:
      - PCIADDRESS
EOF

export KUBECONFIG=/etc/rancher/rke2/rke2.yaml; export KUBECTL=/var/lib/rancher/
rke2/bin/kubectl

```

```

while true; do
    export NODE_NAME=$((${KUBECTL} --kubeconfig=${KUBECONFIG} get nodes -o json
| jq -r '.items[] | select(.metadata.labels."node-role.kubernetes.io/control-plane" !
= "true" and (.status.conditions[]? | select(type=="object") | select(.type=="Ready"
and .status=="True")) | .metadata.name' | head -n1)

    syncStatus=$((${KUBECTL} --kubeconfig=${KUBECONFIG} get
sriovnetworknodestates.sriovnetwork.openshift.io -n sriov-network-operator -ojson | jq -
r ".items[] | select(.metadata.name=="${NODE_NAME}") | .status.syncStatus")
    if [ -z "$syncStatus" ] || [ "$syncStatus" != "Succeeded" ]; then
        sleep 10
    else
        break
    fi
done

input=$((${KUBECTL} --kubeconfig=${KUBECONFIG} get cm sriov-custom-auto-config -n
sriov-network-operator -ojson | jq -r '.data."config.json"')
jq -c '.[.]' <<< $input | while read i; do
    interface=$(echo $i | jq -r '.interface')
    pfname=$(echo $i | jq -r '.pfname')
    resourceName=$(echo $i | jq -r '.resourceName')

    # Check if the resource already exists
    if ${KUBECTL} --kubeconfig=${KUBECONFIG} get
sriovnetworknodepolicy.sriovnetwork.openshift.io "atip-$resourceName" -n sriov-network-
operator &>/dev/null; then
        echo "Resource $resourceName already exists, skipping."
        continue
    fi

    pciaddress=$((${KUBECTL} --kubeconfig=${KUBECONFIG} get
sriovnetworknodestates.sriovnetwork.openshift.io -n sriov-network-operator -ojson |
jq -r ".items[] | select(.metadata.name=="${NODE_NAME}") | .status.interfaces[] |
select(.name=="${interface}") | .pciAddress")
    vendor=$((${KUBECTL} --kubeconfig=${KUBECONFIG} get
sriovnetworknodestates.sriovnetwork.openshift.io -n sriov-network-operator -ojson |
jq -r ".items[] | select(.metadata.name=="${NODE_NAME}") | .status.interfaces[] |
select(.name=="${interface}") | .vendor")
    deviceid=$((${KUBECTL} --kubeconfig=${KUBECONFIG} get
sriovnetworknodestates.sriovnetwork.openshift.io -n sriov-network-operator -ojson |
jq -r ".items[] | select(.metadata.name=="${NODE_NAME}") | .status.interfaces[] |
select(.name=="${interface}") | .deviceID")
    driver=$(echo $i | jq -r '.driver')

    yamlContent=$(sed -e "s/RESOURCENAME/$resourceName/g" \
-e "s/DRIVER/$driver/g" \

```

```

        -e "s/PFNAMES/$pname/g" \
        -e "s/VENDOR/$vendor/g" \
        -e "s/DEVICEID/$deviceid/g" \
        -e "s/PCIADDRESS/$pciaddress/g" \
        -e "s/NUMVF/$(echo $i | jq -r '.numVFsToCreate')/g" /var/sriov-
networkpolicy-template.yaml)

        echo "$yamlContent" | ${KUBECTL}
--kubeconfig=${KUBECONFIG} apply -f -
done

- name: udp-tuning.conf
content: |
net.core.rmem_max = 1342177280
net.core.wmem_max = 516777216
net.core.rmem_default = 10000000
net.core.wmem_default = 10000000
net.core.netdev_max_backlog = 416384
net.core.optmem_max = 25165824
net.ipv4.udp_mem = 11416320 15221760 22832640
net.core.netdev_budget = 1024
net.ipv4.udp_rmem_min = 16384
net.ipv4.udp_wmem_min = 16384
net.ipv4.conf.all.rp_filter = 0
net.core.optmem_max = 4194304
net.ipv4.conf.all.send_redirects = 0

```

10.2 Sample custom file to copy scripts to the file system

```

- name: 100-performance.sh
content: |
#!/bin/bash
# create the folder to extract the artifacts there
mkdir -p /opt/performance-settings
# copy the artifacts
cp performance-settings.sh /opt/performance-settings/

- name: 101-sriov.sh
content: |
- name: 100-performance.sh
content: |
#!/bin/bash
# create the folder to extract the artifacts there
mkdir -p /opt/performance-settings

```

```

# copy the artifacts
cp performance-settings.sh /opt/performance-settings/

- name: 101-sriov.sh
content: |
#!/bin/bash
# create the folder to extract the artifacts there
mkdir -p /opt/sriov
# copy the artifacts
cp sriov-auto-filler.sh /opt/sriov/sriov-auto-filler.sh

- name: 102-udp-tuning.sh
content: |
#!/bin/bash
if ip link show eth2 > /dev/null 2>&1; then
cp udp-tuning.conf /etc/sysctl.d/udp-tuning.conf
sysctl -p /etc/sysctl.d/udp-tuning.conf
sysctl kernel.sched_rt_runtime_us=-1
ethtool -C eth2 rx-usecs 50
ethtool -C eth2 tx-usecs 50
ethtool -C eth2 adaptive-rx on
ethtool -C eth2 adaptive-tx on
ethtool -K eth2 rx-checksum on
ethtool -K eth2 gro on
fi
#!/bin/bash
# create the folder to extract the artifacts there
mkdir -p /opt/sriov
# copy the artifacts
cp sriov-auto-filler.sh /opt/sriov/sriov-auto-filler.sh

- name: 102-udp-tuning.sh
content: |
#!/bin/bash
if ip link show eth2 > /dev/null 2>&1; then
cp udp-tuning.conf /etc/sysctl.d/udp-tuning.conf
sysctl -p /etc/sysctl.d/udp-tuning.conf
sysctl kernel.sched_rt_runtime_us=-1
ethtool -C eth2 rx-usecs 50
ethtool -C eth2 tx-usecs 50
ethtool -C eth2 adaptive-rx on
ethtool -C eth2 adaptive-tx on
ethtool -K eth2 rx-checksum on
ethtool -K eth2 gro on
fi

```

11 Legal notice

Copyright © 2006–2026 SUSE LLC and contributors. All rights reserved.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or (at your option) version 1.3; with the Invariant Section being this copyright notice and license. A copy of the license version 1.2 is included in the section entitled "GNU Free Documentation License".

SUSE, the SUSE logo and YaST are registered trademarks of SUSE LLC in the United States and other countries. For SUSE trademarks, see <https://www.suse.com/company/legal/>.

Linux is a registered trademark of Linus Torvalds. All other names or trademarks mentioned in this document may be trademarks or registered trademarks of their respective owners.

Documents published as part of the series SUSE Technical Reference Documentation have been contributed voluntarily by SUSE employees and third parties. They are meant to serve as examples of how particular actions can be performed. They have been compiled with utmost attention to detail. However, this does not guarantee complete accuracy. SUSE cannot verify that actions described in these documents do what is claimed or whether actions described have unintended consequences. SUSE LLC, its affiliates, the authors, and the translators may not be held liable for possible errors or the consequences thereof.

12 GNU Free Documentation License

Copyright © 2000, 2001, 2002 Free Software Foundation, Inc. 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA. Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition. The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.

- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all

Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>. Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

Copyright (c) YEAR YOUR NAME.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2

```
or any later version published by the Free Software Foundation;  
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.  
A copy of the license is included in the section entitled "GNU  
Free Documentation License".
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “ with... Texts.” line with this:

```
with the Invariant Sections being LIST THEIR TITLES, with the  
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.