HPE Telco Core Validated Reference Design with Robin.io Cloud Native Platform

Test Report

Abstract

This document captures the test cases, execution steps and results of HPE Telco Core Robin platform reference design. This document is intended for the system integration test personnel who customize this reference architecture according to the end customer's requirements and deploy the optimized solution at the customer site or factory and execute the test cases. The reader must have an understanding of the NFV solution and be familiar with the HPE NFV hardware.

Published: June 2020
Edition: 1

Notices

The information contained herein is subject to change without notice. The only warranties for Hewlett Packard Enterprise products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Hewlett Packard Enterprise shall not be liable for technical or editorial errors or omissions contained herein.

Confidential computer software. Valid license from Hewlett Packard Enterprise required for possession, use, or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Links to third-party websites take you outside the Hewlett Packard Enterprise website. Hewlett Packard Enterprise has no control over and is not responsible for information outside the Hewlett Packard Enterprise website.

Acknowledgments

Intel®, Itanium®, Optane™, Pentium®, Xeon®, Intel Inside®, and the Intel Inside logo are trademarks of Intel Corporation in the U.S. and other countries.

Mellanox™ and Mellanox ConnectX™ are trademarks of Mellanox Technologies in the United States and other countries.

Microsoft® and Windows® are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.

Red Hat® is a registered trademark of Red Hat, Inc. in the United States and other countries.

Java® and Oracle® are registered trademarks of Oracle and/or its affiliates.

UNIX® is a registered trademark of The Open Group.

Robin™ is trademark of Robin.io™ in the United States and other countries.

Revision history

Publication date	Edition	Summary of changes
June 2020	1	First version

Contents

Overview

These test cases are planned for Robin platform validation on HPE hardware. The test cases can be run from Robin dashboard or using Robin CLI commands

Assumptions

The testing teams have experience with OpenStack, are trained with the Robin platform and understand tested HW.

Prerequisites

- Robin Installation guide.
- Download OS, DPDK driver and Robin Artifacts version as listed in the appendix.
- Robin platform license

Test Conditions

- Pass – Passed successfully
- Partially passed – Passed with comments (can be correct in a later release or in a roadmap commitment)
- Failed – The test case was not executed successfully

Installation and Configuration Test Cases

Installation

Description	Install Robin Master and Worker roles on physical server
Objective	Verify that the Robin Master and Worker is installed successfully
Prerequisite/s	**1.** Before the installation: **a.** Configure BIOS, Grub settings listed in Appendix **b.** Install OS and drives listed in Appendix **c.** Run the pre-requisite script listed in Appendix and fix errors **2.** Follow the Robin Installation Manual and install the Master. **3.** Follow the Robin Installation Manual and install the Worker.

Test Execution	**Master Node Installation**
	1. Login to the Master node as root user
	2. Run the below command to install Robin on Master Node
	`# ./robin-install-k8s_5.2.4-373.sh server --username=<Robin Username> --password=<Robin Password> --robin-image-archive=robinbinimg-5.2.4-373.tar.gz --k8s-image-archive=k8s-images-5.2.4-373.tar.gz --nics=<nic name>:<vlan>:untagged --ignore-warnings`
	E.g.:
	`# ./robin-install-k8s_5.2.4-373.sh server --username=robin --password=Robin123 --robin-image-archive=robinbinimg-5.2.4-373.tar.gz --k8s-image-archive=k8s-images-5.2.4-373.tar.gz --nics=bond-mgmt:361:untagged --ignore-warnings`
	3. After the successful completion of the above command, the Robin should be installed on Master Node
	Worker Node Installation
	1. Login to the Worker node as root user
	2. Run the below command to install Robin on Worker Node
	`# ./robin-install-k8s_5.2.4-373.sh agent --server=<Robin Master Node IP>  --username=<Robin Username> --password=<Robin Password> --robin-image-archive=robinbinimg-5.2.4-373.tar.gz --k8s-image-archive=k8s-images-5.2.4-373.tar.gz --nics=<nic name>:<vlan>:untagged --ignore-warnings`
	E.g.:
	`# ./robin-install-k8s_5.2.4-373.sh agent --server=xx.xx.xx.xx --username=robin --password=Robin123 --robin-image-archive=robinbinimg-5.2.4-373.tar.gz --k8s-image-archive=k8s-images-5.2.4-373.tar.gz --nics=bond-mgmt:361:untagged --ignore-warnings`

	4. After the successful completion of the above command, the Robin should be installed on Worker Node
Expected Result	**1.** Robin automated verification steps have completed successfully. **a** Automated K8s install + in-built networking + storage discovery **2.** Robin Master is successfully installed. 3. Robin Worker is successfully installed. 4. Robin Dashboard should be accessible - Check that the ROBIN controllers and computes are configured
Status	OK
Comments	Robin pre-checks are completed without any failures in Master and Worker Nodes Robin Master Node installation is completed successfully Robin Worker Node installation is completed successfully Robin Dashboard is accessible from Browser through https://<RobinDashboardIP>:29443

Setting up Robin Cluster

Description	Configure Cluster for Robin to fully operational
Objective	Verify Robin Cluster configuration status
Prerequisite/s	1. Robin Master Node should be installed and running 2. Robin Worker Node should be installed and running

Test Execution	**CLI Robin Login**
	1. Login to Robin Master node
	2. Run below command to enable Robin login in CLI
	`# robin login <Robin Username> --password <Robin Password>`
	3. The command should provide successful login message
	`User <Robin Username>  is logged into Administrators tenant`
	Apply the License for Robin
	1. Login to the Master Node
	2. Run below command to get the license id
	`# robin license id`

	3. Generate the license as per the Robin Manual using the license id from step #2 4. Apply the license # robin license apply <license key> 5. The license should be applied successfully **Resource Pool assignment** **1.** Run below command to add Worker node to default resource pool # robin host assign-rpool <hostnames> default --wait 2. The host should be assigned to default resource pool **Worker Node Role assignment** 1. Run below command to assign compute and storage roles to Worker Node # robin host add-role <hostnames> Compute,Storage –wait 2. The host should be assigned with Compute and Storage roles 3. Run the below command to list the hosts and verify the role associated with the hosts # robin host list **File Collection Creation** 1. Run below command to create file collection # robin collection create <media_type> <rpool_name> --collection-type <type> --replicas <count> --size <size> E.g.: # robin collection create SSD default --collection_type FILE_COLLECTION --replicas 1 --size 5GB --wait 2. File collection should be created successfully
Expected Result	1. Robin license should be applied successfully 2. Robin Worker Node should be assigned with Compute and Storage Roles 3. Robin File Collection should be created successfully
Status	OK

Comments	Robin license is successfully applied on Master Node Robin worker node is successfully added to default resource pool Robin worker node is successfully assigned to Compute and Storage roles Robin file collection is created successfully

Verify Robin Health

Description	Robin Health Check
Objective	Verify that Robin is healthy.
Prerequisite/s	The Robin Node is installed successfully.

Test Execution	**1.** Login to Robin Dashboard from Browser using the below URL and verify the services in Dashboard
	`https://<Robin Dashboard IP>:29443/`
	2. Verify Infrastructure Node status under Dashboard Infrastructure Information Section
	3. Verify Nodes, Storage, Resource Pools and Network status details under Infrastructure Menu Option.
	4. Check Master services status by running the command
	`#  robin host info <master node fqdn>`
	5. Check Worker services status by running the command
	`#  robin host info <worker node fqdn>`
	6. Verify the status of the services of all hosts
	`#  robin host list –services`

Expected Result	1. The Robin Dashboard should be accessible and able to login 2. The Master Node services should be in UP state 3. The Worker Node services should be in UP state 4. The services should not be in Critical state.
Status	OK
Comments	Robin Dashboard login is accessible Robin Dashboard displays infrastructure information including number of nodes, health service status, memory/cpu/storage capacity Robin services are in healthy state

Sample application deployment

Description	Deploy MySQL container on worker node
Objective	Verify sample application deployment with automated storage provisioning
Prerequisite/s	1. Robin Cluster is operational and running 2. Robin Application Bundle for MySQL should be available

Test Execution	1. Login to Master Node
	2. Run the below command to upload the bundle to Robin
	`# robin bundle add <name> <version> <bundle path>`
	`E.g.:`
	`# robin bundle add MySQL 5.7 ./docker-mysql-5.7-333_master.tar.gz`
	3. The bundle should be uploaded successfully
	4. Login to Robin Dashboard
	5. Navigate to Application Bundles
	6. Click on MySQL Bundle
	7. Provide required details and click on “Provision Application”
	8. The application should be provisioned successfully
	9. Login to the application using Application Console

Expected Result	MySQL application is successfully provisioned in Robin Cluster with selected cores, memory, storage and network.
Status	OK
Comments	Application Bundle for MySQL successfully uploaded to Robin MySQL application is successfully provisioned Login to MySQL is successful through application console and database connection is successful.

Network reachability

Description	Deploy PODs with single network
Objective	Verify that PODs can communicate with other PODs in the cluster
Prerequisite/s	1. Robin Cluster is operational and running 2. Application Bundle should be present in Robin

Test Execution	1. Login to Master Node
	2. Run below command to create ip pool based on ovs driver
	`# robin ip-pool add ovsnet1 --ranges <network range> --netmask <netmask> --driver ovs --gateway <network gateway> --vlan <vlanid>`
	E.g.:
	`# robin ip-pool add ovsnet1 --ranges 10.xx.xx.100-110 --netmask 255.255.255.0 --driver ovs --gateway 10.xx.xx.1 –vlan 361`
	3. Create Application Bundle with the network created based on ovs
	4. Application should be deployed successfully
	5. List the application instances
	`# robin instance list`
	6. Login to the application using
	`# rbash <instance container name>`
	7. Verify the ip address details using ifconfig
	8. Verify communication with other instance using ping command
	9. Verify internet connectivity by CURL to “google.com”

Expected Result	1. Network, subnet and ports are created successfully 2. PODs are instantiated successfully and have the expected IP address 3. If the application is deployed as 2 POD, there is connectivity between the 2 PODs 4. The POD is able to reach to internet
Status	OK
Comments	The OVS based ip-pool created successfully Application is deployed successfully with ovs ip-pool created The Application successfully configured with the ip address from the ip pool range. Ping to the other application on the same ovs ip-pool is successful Ping to the network gateway hosted in the switch is successful Curl to www.google.com is successful.

Robin Carrier grade

NUMA Awareness

Description	Test the NUMA Awareness behavior.
Objective	Verify the proper behavior of NUMA awareness.
Prerequisite/s	1. Robin Worker should be enabled with NUMA configurations
Test Execution	1. Login to Master Node 2. Run below command to verify the NUMA configurations of worker node `# robin host info <fqdn of worker node>` 3. Verify the NUMA Topology details discovered for the worker node by Robin
Expected Result	1. The NUMA Topology discovered by Robin should be displayed
Status	OK
Comments	Robin host information for the worker node successfully displays the NUMA topology, huge pages, isolated CPU configured, SRIOV virtual function devices, DPDK configured drivers and devices.

CPU Pinning

Description	CPU Pinning testing.
Objective	Verify that CPU pinning operates as expected. Items checked: - Simple CPU pinning with different number of CPUs (up to the number in NUMA).
Prerequisite	1. Worker Node should be configured with isolated CPU as listed in Grub settings under Appendix section. 2. Application bundle should be available in Robin with CPU cores reservation.
Test execution	1. Login to Robin Dashboard 2. Create Application using the application bundle 3. After the application is provisioned 4. Verify the cores allocated to the application

Expected Result	**1.** The cores allocated to the application is from the same NUMA and from the isolated CPU cores.
Status	OK
Comments	Application created successfully The containers provisioned is assigned to the CPU cores from the isolated CPU list of worker node.

Huge Pages

Description	Huge pages testing.
Objective	Verify that Huge pages are configured as expected. Items checked:
Prerequisite	1. Worker Node should be enabled with 1G Huge pages 2. Application bundle with apps configured with 1G huge pages should be available
Test Execution	1. Login to Robin Dashboard 2. Create Application with the bundle enabled for huge pages 3. The Application should be created successfully 4. Verify Application information to check huge pages allocation. `# robin app info <application name>`
Expected Result	1. The application should be deployed with Huge pages 2. The application information should display the huge page details.
Status	OK
Comments	The Application is successfully provisioned in Robin Cluster Robin command displays successfully the huge pages assigned to the applications The number of 1G huge pages available in the worker node is reflected correctly.

SRIOV

<table>
<tr><td><b>Description</b></td><td>Deploy PODs to use SR-IOV</td></tr>
<tr><td><b>Objective</b></td><td>Verify that PODs deploy on SR-IOV computes and use SR-IOV networks</td></tr>
<tr><td><b>Prerequisite/s</b></td><td>1. Worker Node should be enabled with SR-IOV configurations
2. Application bundle which supports SR-IOV should be present</td></tr>
<tr><td><b>Test Execution</b></td><td>1. Login to Master Node
2. Run below commands to create SR-IOV ip-pools

robin vlan add &lt;vlan1&gt; --wait
robin vlan add &lt;vlan2&gt; --wait
robin ip-pool add &lt;sriov net1&gt; --ranges &lt;net1 range&gt; --netmask &lt;net1 mask&gt; --driver sriov --nictags name=&lt;SR-IOV interface name&gt; --ifcount 1 --vlan &lt;vlan1&gt;
robin ip-pool add &lt;sriov net2&gt; --ranges &lt;net2 range&gt; --netmask &lt;net2 mask&gt; --driver sriov --nictags name=&lt;SR-IOV interface name&gt; --ifcount 1 --vlan &lt;vlan2&gt;

3. Create application from Robin Dashboard using the created SRIOV networks
4. Verify the SR-IOV VFs assignment to the application on Worker Node

ip link show</td></tr>
<tr><td></td><td>5. Connect to application instances from Robin Master Node

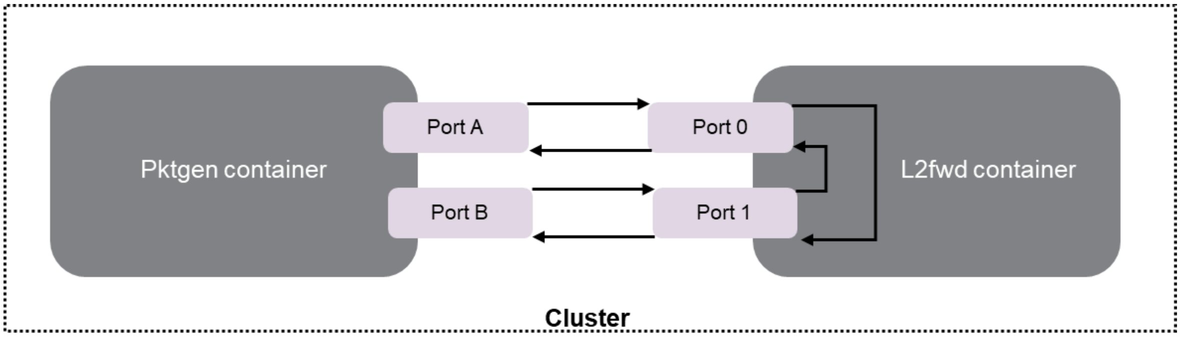
rbash &lt;instance container name&gt;

6. Verify the IP addresses assigned to the container using ifconfig
7. Verify ping to the other instances
8. Verify ping to the external network</td></tr>
<tr><td><b>Expected Result</b></td><td>1. The SR-IOV ip-pools should be created
2. Application should be provisioned with the SR-IOV networks
3. IP addresses should be assigned to the instances
4. Network connectivity should be successful</td></tr>
<tr><td><b>Status</b></td><td>OK</td></tr>
<tr><td><b>Comments</b></td><td><table><tr><td>First executed with NIC</td><td>727055-B21 HPE Ethernet 10Gb 2-port 562SFP+ Adapter</td></tr><tr><td>Test repeated with NIC</td><td>870825-B21 HPE Ethernet 10/25Gb 2-port 661SFP28 Adapter</td></tr></table>
Robin ip-pools with driver SRIOV are created successfully
Application with SRIOV networks is created successfully
The IP addresses are assigned to the instances from created ip-pools
The VFs associated with the SRIOV PODs are visible in the worker node with proper VLAN tags associated.
Ping to the other Pods within the same network is successful
Ping to the gateway is successful</td></tr>
</table>

	Curl to www.google.com is successful

DPDK

Description	Deploy PODs with DPDK
Objective	Verify that the system deploys the POD with DPDK networking.
Prerequisite/s	1. The Worker Node should be enabled with DPDK configurations as mentioned in Appendix 2. SRIOV supported NIC is used in Worker node. Refer to Appendix for hardware BOM. VFs in the NIC will bind with DPDK driver 3. Application Bundle supporting DPDK should be available
Test Execution	1. Login to Master Node 2. Run below commands to create DPDK ip-pools # robin vlan add <vlan1> --wait # robin ip-pool add <dpdk net1> --driver sriov --range <dpdk net1 pool> --prefix 24 --vfdriver igb_uio --nictags=name=<interface1 name> --vlan <vlan1> # robin ip-pool add <dpdk net2> --driver sriov --range <dpdk net2 pool> --prefix 24 --vfdriver igb_uio --nictags=name=<interface2 name> --vlan <vlan1> 3. Create application from Robin Dashboard using the created DPDK networks 4. Connect to l2fwd application instance from Robin Master Node in new terminal # rbash <appname>. l2fwd.01 [l2fwd-01 ~]# /.robin/scripts/cmds.sh # (which display the l2fwd command to execute) [l2fwd-01 ~]# /root/l2fwd/l2fwd -l 10,2,1 -w $ROBIN_MYNET1_PCI_ADDR -w $ROBIN_MYNET2_PCI_ADDR -n 4 --in-memory -- -p 3 # (Modify MYNET1 and MYNET2 to the DPDK network names) # Note down the Port 0 Mac and Port 1 Mac displayed 5. Connect to pktgen application instance from Robin Master Node # rbash <appname>. pktgen.01 [pktgen-01]# ~]# /.robin/scripts/cmds.sh # (which display the pktgen command to execute) [pktgen-01]# ~]# /root/pktgen/pktgen -l 8,7,6,5,4 -w $ROBIN_MYNET1_PCI_ADDR -w $ROBIN_MYNET2_PCI_ADDR -n 4 --in-memory -- -m [8:7].0 -m [6:5].1 #(Modify MYNET1 and MYNET2 to the dpdk network names) set 0 dst mac <l2fwd app Port 0 MAC> set 1 dst mac <l2fwd app Port 1 MAC> set 0,1 proto udp set 0,1 rate 80 set 0,1 size 1024 start 0,1 # Verify the traffic Rx/Tx between the Pktgen and L2fwd instances using DPDK

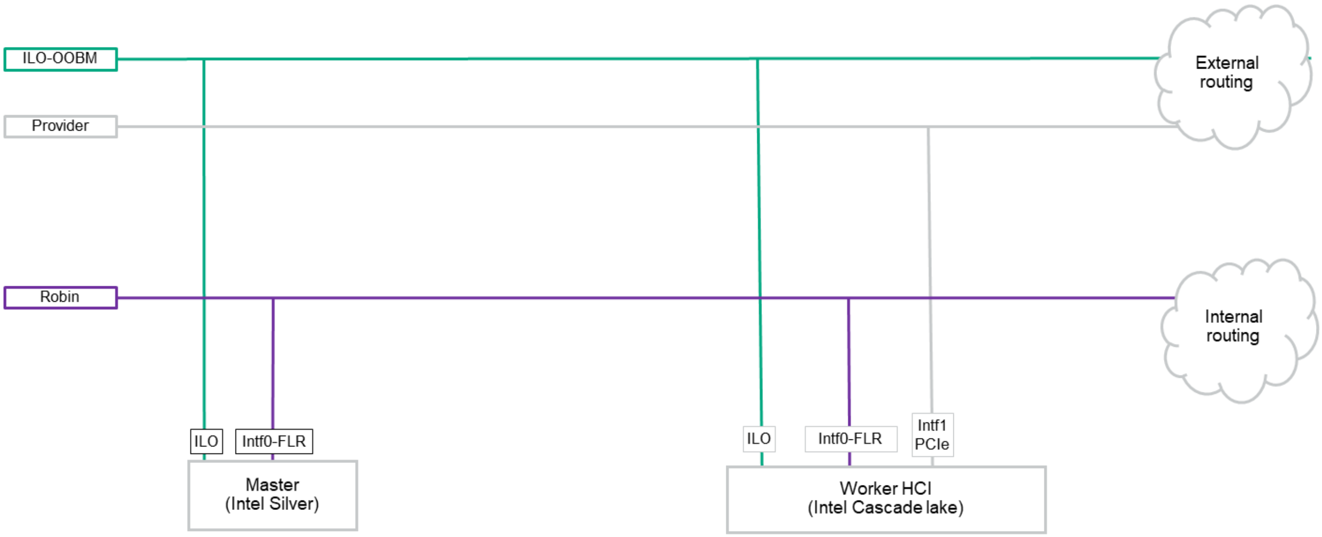
<table>
<tr><td></td><td></td></tr>
<tr><td><strong>Expected Result</strong></td><td>1. IP pools should be created with igb_uio driver
2. Application should be provisioned with DPDK bundle
3. Traffic should flow between pktgen and l2fwd instances.</td></tr>
<tr><td><strong>Status</strong></td><td>OK</td></tr>
<tr><td><strong>Comments</strong></td><td>Robin DPDK IP pools with SRIOV driver and igb_uio VF driver are created successfully
Application with DPDK IP pools is created successfully


<table>
<tr><td>First executed with NIC</td><td>727055-B21 HPE Ethernet 10Gb 2-port 562SFP+ Adapter</td></tr>
<tr><td>Test repeated with NIC</td><td>870825-B21 HPE Ethernet 10/25Gb 2-port 661SFP28 Adapter</td></tr>
</table>
Pktgen is a traffic generator application. Start pktgen app successfully
l2fwd is a layer-2 (L2) forwarding packet processing application using DPDK. It takes traffic from a single RX port and transmits it with few modification on a single TX port.
Configure the same VLAN on both connection pairs:
• VLAN x on PortA and port0
• VLAN x on PortB and port1
Start l2fwd app successfully. The port 0 and port 1 Rx/Tx stats are reflecting with packets count received and transmitted successfully.</td></tr>
</table>

Appendix 1: Worker node hardware BOM

Quantity	Product #	Product Description
1	P19720-B21	HPE ProLiant DL380 Gen10 8SFF NC Configure-to-order Server
1	P02514-L21	HPE DL380 Gen10 Intel Xeon-Gold 6248 (2.5GHz/20-core/150W) FIO Processor Kit
1	P02514-B21	HPE DL380 Gen10 Intel Xeon-Gold 6248 (2.5GHz/20-core/150W) Processor Kit
12	P00924-B21	HPE 32GB (1x32GB) Dual Rank x4 DDR4-2933 CAS-21-21-21 Registered Smart Memory Kit
4	881457-B21	HPE 2.4TB SAS 12G Enterprise 10K SFF (2.5in) SC 3yr Wty 512e Digitally Signed Firmware HDD
1	804331-B21	HPE Smart Array P408i-a SR Gen10 (8 Internal Lanes/2GB Cache) 12G SAS Modular Controller
1	P01366-B21	HPE 96W Smart Storage Battery (up to 20 Devices) with 145mm Cable Kit
1	817749-B21	HPE Ethernet 10/25Gb 2-port 640FLR-SFP28 Adapter
1	870825-B21	HPE Ethernet 10/25Gb 2-port 661SFP28 Adapter
1	727055-B21	HPE Ethernet 10Gb 2-port 562SFP+ Adapter
2	865408-B21	HPE 500W Flex Slot Platinum Hot Plug Low Halogen Power Supply Kit

Appendix 2: Deployment diagram



Appendix 3: Configuration parameters

BIOS settings

The BIOS settings should be enabled for Virtualization, NUMA and SRIOV.

"ProcVirtualization": "Enabled"
"SubNumaClustering": "Disabled"
"Sriov": "Enabled"
"NodeInterleaving": "Disabled"

Grub settings

- Parameters:

```
GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR="$(sed 's, release .*$,,g' /etc/system-release)"
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
GRUB_TERMINAL_OUTPUT="console"
GRUB_CMDLINE_LINUX="crashkernel=auto spectre_v2=retpoline
rd.lvm.lv=centos/root rd.lvm.lv=centos/swap rhgb quiet iommu=pt
intel_iommu=on hugepagesz=1G hugepages=50 isolcpus=1-10"
GRUB_DISABLE_RECOVERY="true"
```

- NUMA awareness:

```
isolcpus=1-10
```

Isolation of CPUs: The options will instruct the kernel not to schedule any process on these vCPUs without explicit pinning.

- Huge Pages:

```
hugepagesz=1G hugepages=50
```

The options in grub will allocate 50G of huge pages of size 1G.

- DPDK:

```
DPDK 20.05
```

DPDK version on the host. Bind VF in the NIC to DPDK driver.

- SRIOV/DPDK:

SR-IOV requires support in the BIOS as well as in the operating system instance or hypervisor that is running on the hardware NIC naming

```
[root@pocf02 ~]# cat /etc/udev/rules.d/70-persistent-net.rules
ACTION=="add", SUBSYSTEM=="net", ATTR{address}=="00:25:90:fa:7b:4f",
ATTR{type}=="1", NAME:="sriov0"
ACTION=="add", ...
```

Configuring Virtual functions:

```
[root@pocf02 ~]# cat /etc/udev/rules.d/75-allocate-sriov-vfs.rules
ACTION=="move", SUBSYSTEM=="net", KERNEL=="sriov[0-9]*", RUN+="/bin/sh
-c '/usr/bin/echo 4 > /sys/class/net/%k/device/sriov_numvfs'"
ACTION=="change", SUBSYSTEM=="net", KERNEL=="sriov[0-9]*",
RUN+="/bin/sh
-c '/usr/bin/echo 4 > /sys/class/net/%k/device/sriov_numvfs'"
```

Verifying Virtual functions:

```
[root@pocf02 ~]# ip link show sriov0
3: sriov0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP
mode DEFAULT group default qlen 1000
link/ether 00:25:90:fa:7b:4f brd ff:ff:ff:ff:ff:ff
vf 0 MAC 9e:62:16:73:78:c2, spoof checking on, link-state auto,
trust off, query_rss off
vf 1 MAC de:ed:94:72:47:f2, spoof checking on, link-state auto,
trust off, query_rss off
vf 2 MAC d2:fe:8f:2f:d2:d0, spoof checking on, link-state auto,
trust off, query_rss off
vf 3 MAC 00:00:00:00:00:00, spoof checking on, link-state auto,
trust off, query_rss off
```

Security settings

- Disable SELinux
- Disable Firewall

Appendix 4: Artifact Files

OS Image

In the ATP, Cent OS is used as OS:

- Centos 7.7 kernel 3.10.0-1062

Robin Images

Pre-installation script to validate the host settings.

Note: Pre-install script output errors are fixed. Warnings can be ignored

In the ATP, 3 images were used. They are:

- Robin installer scripts
- Kubernetes images
- Robin application as container images

Application Bundle

- MySQL Bundle - docker-mysql-5.7-333_master.tar.gz
- DPDK application Bundle - docker-dpdk-v2-517_master.tar.gz

Drivers and compiled image

The files names are listed below:

i40e Driver Version	2.7.29
i40evf Driver version	3.6.15
DPDK Version	20.05