

Agilio SmartNIC vRouter Getting Started Guide

Proprietary and Confidential

2017-03-07

Contents

1	Revision History	4
2	Overview	5
3	Installation and Setup	6
3.1	Supported Platforms	6
3.2	Host Requirements	6
3.3	Agilio vRouter Packaging	6
3.4	Installing Agilio Vrouter	6
3.4.1	Fabric	7
3.4.1.1	Part One	7
3.4.1.2	Agilio vRouter Fabric Testbed Definitions	7
3.4.1.3	Part Two	8
3.4.1.4	Installing via Fabric - Quick steps (Reference Only)	9
3.5	Additional Fabric Tasks	10
3.5.1	Upgrading Contrail Software	10
3.5.1.1	Standard Procedure	10
3.5.1.2	Additional Steps when using SmartNIC	11
3.5.1.3	Upgrading Contrail Software - Quick Steps (Reference Only)	12
3.5.2	Uninstalling Contrail	12
3.5.3	Uninstalling Agilio SmartNIC	12
3.5.4	Adding a compute node to an existing cluster	13
3.5.5	Removing a compute node from an existing cluster	13
4	Testbed example files	14
4.1	Two Accelerated SmartNIC Compute Example	14
4.2	One Accelerated and One Unaccelerated SmartNIC Compute Example	16

5	Configuring OpenStack for Acceleration	18
5.1	Accelerating interfaces with virtio	18
5.2	Accelerating interfaces with SR-IOV	18
6	Appendix	19
6.1	Contrail Links and Documents	19
6.2	Netronome Links and Documents	19
6.3	PCIe Quirk Kernel Patch	19

COPYRIGHT

No part of this publication or documentation accompanying this Product may be reproduced in any form or by any means or used to make any derivative work by any means including but not limited to by translation, transformation or adaptation without permission from Netronome Systems, Inc., as stipulated by the United States Copyright Act of 1976. Contents are subject to change without prior notice.

WARRANTY

Netronome warrants that any media on which this documentation is provided will be free from defects in materials and workmanship under normal use for a period of ninety (90) days from the date of shipment. If any defect in any such media should occur during this 90-day period, the media may be returned to Netronome for a replacement.

NETRONOME DOES NOT WARRANT THAT THE DOCUMENTATION SHALL BE ERROR-FREE. THIS LIMITED WARRANTY SHALL NOT APPLY IF THE DOCUMENTATION OR MEDIA HAS BEEN (I) ALTERED OR MODIFIED; (II) SUBJECTED TO NEGLIGENCE, COMPUTER OR ELECTRICAL MALFUNCTION; OR (III) USED, ADJUSTED, OR INSTALLED OTHER THAN IN ACCORDANCE WITH INSTRUCTIONS FURNISHED BY NETRONOME OR IN AN ENVIRONMENT OTHER THAN THAT INTENDED OR RECOMMENDED BY NETRONOME.

EXCEPT FOR WARRANTIES SPECIFICALLY STATED IN THIS SECTION, NETRONOME HEREBY DISCLAIMS ALL EXPRESS OR IMPLIED WARRANTIES OF ANY KIND, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT AND FITNESS FOR A PARTICULAR PURPOSE.

Some jurisdictions do not allow the exclusion of implied warranties, so the above exclusion may not apply to some users of this documentation. This limited warranty gives users of this documentation specific legal rights, and users of this documentation may also have other rights which vary from jurisdiction to jurisdiction.

LIABILITY

Regardless of the form of any claim or action, Netronome's total liability to any user of this documentation for all occurrences combined, for claims, costs, damages or liability based on any cause whatsoever and arising from or in connection with this documentation shall not exceed the purchase price (without interest) paid by such user.

IN NO EVENT SHALL NETRONOME OR ANYONE ELSE WHO HAS BEEN INVOLVED IN THE CREATION, PRODUCTION, OR DELIVERY OF THE DOCUMENTATION BE LIABLE FOR ANY LOSS OF DATA, LOSS OF PROFITS OR LOSS OF USE OF THE DOCUMENTATION OR FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, EXEMPLARY, PUNITIVE, MULTIPLE OR OTHER DAMAGES, ARISING FROM OR IN CONNECTION WITH THE DOCUMENTATION EVEN IF NETRONOME HAS BEEN MADE AWARE OF THE POSSIBILITY OF SUCH DAMAGES. IN NO EVENT SHALL NETRONOME OR ANYONE ELSE WHO HAS BEEN INVOLVED IN THE CREATION, PRODUCTION, OR DELIVERY OF THE DOCUMENTATION BE LIABLE TO ANYONE FOR ANY CLAIMS, COSTS, DAMAGES OR LIABILITIES CAUSED BY IMPROPER USE OF THE DOCUMENTATION OR USE WHERE ANY PARTY HAS SUBSTITUTED PROCEDURES NOT SPECIFIED BY NETRONOME.

1 Revision History

Date	Revision	Author	Description
2017-02-24	0.1	M.Neiderhauser	Initial commit of the document
2017-02-28	0.2	M.Neiderhauser	Update for JTAC
2017-03-06	0.3	M.Bridger	Review
2017-03-07	0.4	M.Neiderhauser	Added <code>uninstall_ns_agilio_nic</code> . Updated overview

2 Overview

This getting started guide documents the fabric based install procedure for installing Netronome Agilio vRouter using Netronome Agilio SmartNICs.

The fabric based install procedure for Netronome Agilio vRouter is applicable for the Netronome Agilio SmartNICs listed in [Supported Platforms](#). Netronome Agilio SmartNICs can be configured in accelerated and non-accelerated modes. All steps in this document should be followed for SmartNICs in either mode.

The fabric `testbed.py` file specifies which nodes contains SmartNICs and which nodes should be accelerated. For more details on the `testbed.py` file please see the [Agilio vRouter Fabric Testbed Definitions](#) section.

Examples of the `testbed.py` file can be found in the [Testbed example files](#) section for SmartNICs in both accelerated and non-accelerated modes.

This document covers the following tasks:

- 1) [Installing Agilio vRouter](#)
 - a) Installing Contrail packages
 - b) [Creating and populating the testbed file](#)
 - c) Upgrading the kernel
 - d) Installing Agilio vRouter packages (SmartNIC packages)
 - e) Setting SmartNIC media settings
 - f) `fab install_contrail`
 - g) `fab setup_interface`
 - h) `fab setup_all`
- 2) [Upgrading Contrail software](#)
- 3) [Upgrading Agilio vRouter software](#)
- 4) [Uninstalling Contrail](#)
- 5) [Uninstalling Agilio SmartNIC](#)
- 6) [Adding compute nodes](#)
- 7) [Removing compute nodes](#)
- 8) [Configuring OpenStack for acceleration](#)

For details on any other fabric or Contrail task please review the Contrail Getting Started Guide:

https://www.juniper.net/techpubs/en_US/contrail3.1/information-products/pathway-pages/getting-started-pwp.html

Contrail vRouter requirements and limitations are imposed on Netronome Agilio vRouter requirements and limitations.

This document provides useful links to both Contrail specific resources and Netronome specific resources.

3 Installation and Setup

3.1 Supported Platforms

The Agilio vRouter software is supported on the Netronome Intelligent Server Adapter or SmartNIC. The following Netronome Intelligent Server Adapters are supported by this release of Agilio vRouter:

- 2x10G Agilio-CX - Half height, half length. (AMDA-0096-0001 / AMDA-0099-0001)
- 1x40G Agilio-CX - Half height, half length. (AMDA-0081-0001)
- 2x40G Agilio-CX - Half height, half length. (AMDA-0097-0001)
- 2x25G Agilio-CX - Half height, half length. (AMDA-0099-0001)
- 4x10G Breakout cables supported on each of the adapters supporting 40G interfaces.

3.2 Host Requirements

The Agilio vRouter software is compatible with the following distributions:

- Ubuntu 14.04.5 with minimal Kernel version of 3.13.0-100
- Contrail 3.1.3.0
- Openstack Kilo / Mitaka

As of writing this guide Contrail vRouter Fabric will automatically install the 3.13.0-106 Kernel.

3.3 Agilio vRouter Packaging

The debian package named `ns-agilio-vrouter-depends-packages_x.x.x.x-xxx_amd64.deb` is a supplemental package to the `contrail-install-packages-x.x.x.x-xxx-openstack_version_all.deb` provided by Juniper. The `ns-agilio-vrouter-depends-packages_x.x.x.x-xxx_amd64.deb` contains everything needed to install and configure a vRouter cluster with an Agilio SmartNIC. This includes but is not limited to the following:

- Netronome Agilio CoreNIC package
- Netronome Board Support package
- Netronome Agilio vRouter package
- Static package repository and setup script
 - Used to provide apt-get functionality for all packages and dependencies
- Netronome built QEMU 2.5 packages

3.4 Installing Agilio Vrouter

The Agilio vRouter software is distributed as a Debian Package.

It is important to know and understand the existing Contrail vRouter installation procedure. Please review the Contrail Getting Started Guide:

https://www.juniper.net/techpubs/en_US/contrail3.1/information-products/pathway-pages/getting-started-pwp.html

It is expected that you already have a compatible Ubuntu operating system installed before installing the Contrail and Netronome packages. Please refer to the **Host Requirements section for minimum versions.**

3.4.1 Fabric

3.4.1.1 Part One

- 1) Ensure that a compatible base operating system has been installed, using the installation instructions for that system.
- 2) Download the appropriate Contrail install packages file from <http://www.juniper.net/support/downloads/?p=contrail#sw>:
 Ubuntu: `contrail-install-packages-x.x.x.x-xxx~openstack_version_all.deb`
- 3) Copy the downloaded Contrail install packages file to `/tmp/` on the first server for your system installation.
- 4) On one of the config nodes in your cluster, copy the Contrail packages as follows:
 Ubuntu: `scp <id@server>:/path/to/contrail-install-packages-x.x.x.x-xxx~openstack_version_all.deb /tmp`
- 5) Install the Contrail packages:
 Ubuntu: `dpkg -i /tmp/contrail-install-packages-x.x.x.x-xxx~openstack_version_all.deb`
- 6) Run the `setup.sh` script. This step creates the Contrail packages repository as well as the Fabric utilities (located in `/opt/contrail/utils`) needed for provisioning:
`cd /opt/contrail/contrail_packages`
`./setup.sh`
- 7) Populate the `testbed.py` definitions file, see [Agilio vRouter Fabric Testbed Definitions](#).

3.4.1.2 Agilio vRouter Fabric Testbed Definitions Please review the following Contrail documents for setting up the Testbed file.

- https://www.juniper.net/techpubs/en_US/contrail3.1/topics/task/installation/testbed-file-vnc.html

Agilio vRouter makes use of the `control_data` dictionary (required) and the `bond` dictionary (required only when SmartNIC is bonded) in the testbed file.

Note: Agilio vRouter currently only supports bond modes 802.1ad and balanced-xor with layer3+4 hashing

Agilio vRouter also requires definition of the `env.ns_agilio_vrouter` dictionary. Please provide appropriate values for the compute nodes in this cluster.

```
env.ns_agilio_vrouter = {
    accel_compute_01: {'huge_page_alloc': '0G', 'huge_page_size': '1G', 'coremask': '2,4',
                       'pinning_mode': 'auto:balance'},
    accel_compute_02: {'huge_page_alloc': '0G', 'huge_page_size': '1G', 'coremask': '2,4',
                       'pinning_mode': 'auto:balance'},
}
```

- `huge_page_alloc`
 (virtiorelayd) Amount of memory to allocate to huge pages for DPDK (e.g., 512M, 64G, ...)

This must be tuned on a system-by-system basis.

The default value of “0G” will cause virtiorelayd startup to exit with an error.

Note: Please see the Agilio SmartNIC vRouter Operational Manual for additional details.

- `huge_page_size`
(virtiorelayd) Size of huge pages to allocate
Note: Please see the Agilio SmartNIC vRouter Operational Manual for additional details.
- `coremask`
(virtiorelayd) CPUs to use for worker threads (either comma separated integers or hex bitmap starting with 0x)
Note: Please see the Agilio SmartNIC vRouter Operational Manual for additional details.
- `pinning_mode`
(virtiorelayd) Fine-grained control over VF to CPU assignments. This can be one of the following:
 - 1) The string “auto:combine”, meaning to pin each VF to a specific single CPU, based on VIRTIORELAYD_CPU_MASK.
 - 2) The string “auto:balance”, meaning to pin each VF to a specific pair of CPUs (split between the VM and VF side of the relay), based on VIRTIORELAYD_CPU_MASK.
 - 3) The string “auto:split”. This is similar to “auto:balance” but allocates CPUs in a way that can produce a lopsided work distribution.
 - 4) A semicolon-delimited list of '<vf>:<cpu>[,<cpu>]' strings specifying exactly which CPU(s) to use for the specified VFs. Note that since this list is semicolon-delimited, it should be enclosed in quotes to prevent the shell from interpreting the semicolons as statement separators.
 - 5) A blank string, meaning to use the automatic CPU scheduling policy (no pins).**Note: Please see the Agilio SmartNIC vRouter Operational Manual for additional details.**
- (optional) `log_level`
(virtiorelayd) Log threshold 0-7 (least to most verbose)
Note: Please see the Agilio SmartNIC vRouter Operational Manual for additional details.

3.4.1.3 Part Two Use Fabric to copy and install Contail packages on the remaining machines in your cluster as follows:

- 1) Ensure that the `testbed.py` file has been created and populated with information specific to your cluster at `/opt/contrail/utils/fabfile/testbeds`.
See Agilio vRouter Fabric Testbed Definitions.
- 2) Run Fabric commands to install packages as follows:
`fab install_pkg_all:/tmp/contrail-install-packages-x.x.x.x-xxx~openstack_version_all.deb`
Note: Fab commands are always run from /opt/contrail/utils
- 3) It is recommended to always use the Contrail-recommended kernel version for Ubuntu-based systems. To get the recommended kernel versions installed, use the following fabric-utils command:
`fab upgrade_kernel_all`
Note: This step installs the recommended kernel version, as an upgrade or as a downgrade, in all nodes and performs reboot. Reconnect to perform remaining tasks.
- 4) Download the appropriate Agilio vRouter Depends Packages file:
Ubuntu: `ns-agilio-vrouter-depends-packages_x.x.x.x-xxx_amd64.deb`

- 5) On one of the config nodes in your cluster, copy the Agilio vRouter Depends Packages as follows:

```
Ubuntu: scp <id@server>:/path/to/ns-agilio-vrouter-depends-packages_x.x.x.x-xxx_amd64.deb /tmp
```

- 6) Deploy and install the Agilio vRouter Depends Packages to the SmartNIC compute nodes.

```
fab install_ns_agilio_nic:/tmp/ns-agilio-vrouter-depends-packages_x.x.x.x-xxx_amd64.deb
```

Note: As in step 1: Ensure that the testbed.py file has properly been populated.

Note: The compute nodes with SmartNICs (accel or non-accel) may be rebooted during this step.

- 7) (optional) SmartNIC media configuration

If it is desired to change the media configuration of the SmartNIC please do so at this time using the nfp-media tool located in /opt/netronome/bin.

```
/opt/netronome/bin/nfp-media -h
```

Example of setting a 1x40G to 4x10G breakout mode

- a) /opt/netronome/bin/nfp-media --set-media=phy0=4x10G
- b) service ns-core-nic.autorun clean
- c) reboot

Note: Media configuration will persist on reboot, powercycles, and re-installs.

- 8) After reboot, install the required Contrail packages in each node of the cluster:

```
fab install_contrail
```

- 9) Since the use of control_data in the testbed file is required, run the following to configure the interfaces:

```
fab setup_interface
```

- 10) Provision the cluster:

```
fab setup_all
```

3.4.1.4 Installing via Fabric - Quick steps (Reference Only) Below is a list of notes and commands for installing Agilio vRouter via Fabric

```
# Copy the contrail-install-packages to /tmp of the cfgm node
```

```
ls /tmp
```

```
# Install the contrail-install-packages on the cfgm node
```

```
dpkg -i /tmp/contrail-install-packages-x.x.x.x-xxx-openstack_version_all.deb
```

```
cd /opt/contrail/contrail_packages && ./setup.sh
```

```
apt-get update
```

```
# Populate /opt/contrail/utils/fabfile/testbeds/testbed.py
```

```
## MANUAL STEP
```

```
# Install contrail-install-packages on remaining nodes
```

```
cd /opt/contrail/utils
```

```
fab install_pkg_all:/tmp/contrail-install-packages-x.x.x.x-xxx-openstack_version_all.deb
```

```
# Upgrade all nodes to recommended kernel
```

```
fab upgrade_kernel_all
```

```
# Reconnect to cfgm node to continue installing the cluster
# Copy the ns-agilio-vrouter-depends-packages to /tmp of the cfgm node
ls /tmp

# Install ns-agilio-vrouter-depends-packages on SmartNIC compute nodes
fab install_ns_agilio_nic:/tmp/ns-agilio-vrouter-depends-packages_x.x.x.x-xxx_amd64.deb

# Install Contrail packages on all nodes
fab install_contrail

# Setup control_data interfaces
fab setup_interface

# Provision the cluster
fab setup_all
```

3.5 Additional Fabric Tasks

3.5.1 Upgrading Contrail Software

NOTE: Before upgrading, ensure VM's are shutdown and are not set to auto boot.

Upgrading Contrail and Agilio vRouter is very similar to the current documentation for vrouter with contrail-vrouter-dkms:

https://www.juniper.net/techpubs/en_US/contrail3.1/topics/task/installation/software-upgrading-cent-ubunt-3.0.html

This guide will ignore step 7 because it is for Contrail Storage options.

The **Standard Procedure** below will supply the upgrade steps from [Contrail's Documentation](#).

The **Additional Steps when using SmartNIC** must **ALWAYS** be ran after running through the **Standard Procedure**, even if SmartNIC components do not have an update.

The **Additional Steps when using SmartNIC** can be ran without going through the **Standard Procedure** to only upgrade SmartNIC components.

3.5.1.1 Standard Procedure

- 1) Download the appropriate Contrail install packages file from <http://www.juniper.net/support/downloads/?p=contrail#sw>:
Ubuntu: `contrail-install-packages-x.x.x.x-xxx-openstack_version_all.deb`
- 2) Copy the downloaded Contrail install packages file to /tmp/ on the first server for your system installation.
- 3) On one of the config nodes in your cluster, copy the Contrail packages as follows:
Ubuntu: `scp <id@server>:/path/to/contrail-install-packages-x.x.x.x-xxx-openstack_version_all.deb /tmp/`
- 4) Install the Contrail packages:
Ubuntu: `dpkg -i /tmp/contrail-install-packages-x.x.x.x-xxx-openstack_version_all.deb`

- 5) Run the `setup.sh` script. This step creates the Contrail packages repository as well as the Fabric utilities (located in `/opt/contrail/utils`) needed for provisioning:

```
cd /opt/contrail/contrail_packages
./setup.sh
```

- 6) Populate the `testbed.py` definitions file, see [Agilio vRouter Fabric Testbed Definitions](#).
- 7) (Optional) There is an option to minimize upgrade down time by dropping the Contrail Analytics keypace before the upgrade, by issuing the following `fab` command:

```
fab drop_analytics_keyspace
```

- 8) Use the following upgrade procedure for Contrail systems based on Ubuntu 14.04 with `contrail-vrouter-dkms` installed. The command sequence upgrades the kernel version and also reboots the compute nodes when finished.

```
fab upgrade_contrail: <from>,/tmp/contrail-install-packages-x.x.x.x-xxx~openstack_version_all.deb
```

NOTE `<from>` refers to the currently installed release number, for example 3.0.2.0, 3.1.2.0, etc.

NOTE Running `fab upgrade_contrail` may fail when the `fab` task attempts to unload and load the `vrouter` kernel module when SmartNIC acceleration is enabled. This is due to a kernel module dependency between the `nfp_vrouter` kernel module (SmartNIC acceleration module) and the `vrouter` kernel module. It is safe to ignore this failure as it will be covered by the section [Additional Steps when using SmartNIC](#).

- 9) (Recommended) All nodes in the cluster can be upgraded to the default kernel version using the following command:

```
fab upgrade_kernel_all
```

Note: This step installs the recommended kernel version, as an upgrade or as a downgrade, in all nodes and performs a reboot. Reconnect to perform remaining tasks.

Note: There may be some implications with the auto install of `vrouter` and `nfp_vrouter dkms` drivers for the upgraded kernel.

3.5.1.2 Additional Steps when using SmartNIC

- 1) Ensure the `testbed.py` file is properly populated
- 2) Download the appropriate Agilio vRouter Depends Packages file:
Ubuntu: `ns-agilio-vrouter-depends-packages_x.x.x.x-xxx_amd64.deb`
- 3) On one of the config nodes in your cluster, copy the Agilio vRouter Depends Packages as follows:
Ubuntu: `scp <id@server>:/path/to/ns-agilio-vrouter-depends-packages_x.x.x.x-xxx_amd64.deb /tmp`

- 4) Deploy and upgrade the Agilio vRouter Depends Packages to the SmartNIC compute nodes.

```
fab upgrade_ns_agilio_contrail:<from>,/tmp/ns-agilio-vrouter-depends-packages_x.x.x.x-xxx_amd64.deb
```

Note: As in step 1: Ensure that the `testbed.py` file has properly populated.

Note: The compute nodes with SmartNICs (accel or non-accel) may be rebooted during this step.

3.5.1.3 Upgrading Contrail Software - Quick Steps (Reference Only) Below is a list of notes and commands for Upgrading Contrail Software including Agilio vRouter via Fabric.

```
# Copy the contrail-install-packages to /tmp of the cfm node
ls /tmp

# Install the contrail-install-packages on the cfm node
dpkg -i /tmp/contrail-install-packages-x.x.x-xxx-openstack_version_all.deb
cd /opt/contrail/contrail_packages && ./setup.sh
apt-get update

# Populate / Verify file: /opt/contrail/utls/fabfile/testbeds/testbed.py
## MANUAL STEP

# OPTIONAL
# fab drop_analytics_keyspace

# Upgrade contrail-install-packages on remaining nodes
# Ex: <from> is the currently installed Contrail version, e.g. 3.1.2.0
# NOTE: Unloading vrouters.ko may fail in upgrade_contrail. Ignore this failure.
cd /opt/contrail/utls
fab upgrade_contrail:<from>,/tmp/contrail-install-packages-x.x.x-xxx-openstack_version_all.deb

# Upgrade all nodes to recommended kernel
## Note this may have some implications with vrouters and nfp_vrouters dkms drivers
fab upgrade_kernel_all

# Reconnect to cfm node to continue installing the cluster
# Copy the ns-agilio-vrouter-depends-packages to /tmp of the cfm node
ls /tmp

# Upgrade ns-agilio-vrouter-depends-packages on SmartNIC compute nodes
# Ex: <from> is the currently installed Contrail version, e.g. 3.1.2.0
fab upgrade_ns_agilio_contrail:<from>,/tmp/ns-agilio-vrouter-depends-packages_x.x.x-xxx_amd64.deb
```

3.5.2 Uninstalling Contrail

- 1) Run the following fabric command to uninstall contrail and acceleration with Agilio SmartNIC

```
fab uninstall_contrail
```

3.5.3 Uninstalling Agilio SmartNIC

- 1) Run the following fabric command to uninstall Agilio SmartNIC configuration, packages, and dependencies

```
fab uninstall_ns_agilio_nic
```

Note: This will remove everything installed by the `fab install_ns_agilio_nic` task with the exception of uninstalling packages `python` and `python-dev`. Kernel command line parameters are not reverted in case additional hardware relies on those parameters.

3.5.4 Adding a compute node to an existing cluster

Please refer to the link below for full instructions on adding a compute node. The steps below are supplemented with Netronome Specific steps.

https://www.juniper.net/techpubs/en_US/contrail3.1/topics/task/installation/add-new-compute-node-vnc.html

- 1) Add the new information about the new compute node(s) into your existing `testbed.py` file.
 - 2) Copy the `contrail-install-packages` file to the `/tmp` directory of the `cfgm` node where the `fab` commands are triggered:
Ubuntu: `scp <id@server>:/path/to/contrail-install-packages-x.x.x.x-xxx~openstack_version_all.deb /tmp`
 - 3) Install the `contrail-install-packages` on to the new compute node (or nodes):
Ubuntu:
`fab install_pkg_node:/tmp/contrail-install-packages-x.x.x.x-xxx~openstack_version_all.deb, host@10.1.1.1`
 - 4) Upgrade the kernel on the new compute node (or nodes):
Ubuntu: `fab upgrade_kernel_node:host@10.1.1.1`
 - 5) On one of the config nodes in your cluster, copy the Agilio vRouter Depends Packages as follows:
Ubuntu: `scp <id@server>:/path/to/ns-agilio-vrouter-depends-packages_x.x.x.x-xxx_amd64.deb /tmp`
 - 6) Deploy and install the Agilio vRouter Depends Packages to the SmartNIC compute nodes.
`fab install_ns_agilio_nic:/tmp/ns-agilio-vrouter-depends-packages_x.x.x.x-xxx_amd64.deb`
 - 7) (optional) SmartNIC media configuration
If it is desired to change the media configuration of the SmartNIC please do so at this time using the `nfp-media` tool located in `/opt/netronome/bin`.
`/opt/netronome/bin/nfp-media -h`
Example of setting a 1x40G to 4x10G breakout mode
 - a) `/opt/netronome/bin/nfp-media --set-media=phy0=4x10G`
 - b) `service ns-core-nic.autorun clean`
 - c) `reboot`
- Note: Media configuration will persist on reboot, powercycles, and re-installs.**
- 8) Use `fab` commands to add the new compute node (or nodes):
Ubuntu: `fab add_vrouter_node:host@10.1.1.1`

3.5.5 Removing a compute node from an existing cluster

Please refer to the link below for full instructions on removing a compute node:

https://www.juniper.net/techpubs/en_US/contrail3.1/topics/task/installation/add-new-compute-node-vnc.html

- 1) `fab detach_vrouter_node:root@10.1.1.1`
- 2) Remove the information about this detached compute node from the existing `testbed.py` file.

4 Testbed example files

4.1 Two Accelerated SmartNIC Compute Example

node_ctrl_0 is a multi role node using onboard 1G NICs.

node_comp_0 is a compute node using a Netronome SmartNIC in 1x40G mode. This node is used for acceleration.

node_comp_1 is a compute node using a Netronome SmartNIC in 4x10G Breakout mode. This node is used for acceleration.

File: /opt/contrail/utils/fabfile/testbeds/testbed.py

```
from fabric.api import env

#Management ip addresses of hosts in the cluster
node_ctrl_0 = 'root@172.18.47.35' # control node - no SmartNIC
node_comp_0 = 'root@172.18.47.39' # compute node - SmartNIC
node_comp_1 = 'root@172.18.47.37' # compute node - SmartNIC

#External routers if any
#for eg.
#ext_routers = [('mx1', '10.204.216.253')]
ext_routers = []

#Autonomous system number
router_asn = 64525

#Host from which the fab commands are triggered to install and provision
host_build = node_ctrl_0

#Role definition of the hosts.
env.roldefs = {
    'all': [node_ctrl_0, node_comp_0, node_comp_1],
    'cfgm': [node_ctrl_0],
    'openstack': [node_ctrl_0],
    'control': [node_ctrl_0],
    'compute': [node_comp_0, node_comp_1],
    'collector': [node_ctrl_0],
    'webui': [node_ctrl_0],
    'database': [node_ctrl_0],
    'build': [host_build],
    'storage-master': [node_ctrl_0],
    'storage-compute': [node_comp_0, node_comp_1],
}

# required if using env.ns_agilio_vrouter
control_data = {
    node_ctrl_0 : { 'ip': '172.18.48.32/24',
                    'gw': '172.18.48.1',
                    'device': 'eth1'},
    node_comp_0 : { 'ip': '172.18.48.33/24',
                    'gw': '172.18.48.1',
                    'device': 'nfp_p0'},
```

```

    node_comp_1 : { 'ip': '172.18.48.34/24',
                    'gw': '172.18.48.1',
                    'device': 'bond0' },
}

# Setup Netronome Agilio vRouter on specified nodes
# Populate to match compute node machine properties
env.ns_agilio_vrouter = {
    node_comp_0: {'huge_page_alloc': '24G',
                  'huge_page_size': '1G',
                  'coremask': '2,4',
                  'pinning_mode': 'auto:split'},
    node_comp_1: {'huge_page_alloc': '24G',
                  'huge_page_size': '1G',
                  'coremask': '2,4',
                  'pinning_mode': 'auto:split'}
}

bond= {
    # 4x10G breakout mode on a 1x40G SmartNIC
    node_comp_1 : { 'name': 'bond0',
                    'member': ['nfp_p0', 'nfp_p1', 'nfp_p2', 'nfp_p3'],
                    'mode': '802.3ad',
                    'xmit_hash_policy': 'layer3+4' },
}

env.hostnames = {
    node_ctrl_0: 'node-ctrl1',
    node_comp_0: 'node-comp-0',
    node_comp_1: 'node-comp-1',
}

# Openstack admin password
env.openstack_admin_password = 'netronome'

# Passwords of each host
env.passwords = {
    node_ctrl_0: 'netronome',
    node_comp_0: 'netronome',
    node_comp_1: 'netronome',
    host_build: 'netronome',
}

# For reimage purpose
env.ostypes = {
    node_ctrl_0: 'ubuntu',
    node_comp_0: 'ubuntu',
    node_comp_1: 'ubuntu',
}

```

4.2 One Accelerated and One Unaccelerated SmartNIC Compute Example

node_ctrl_0 is a multi role node using onboard 1G NICs.

node_comp_0 is a compute node using a Netronome SmartNIC in 1x40G mode. This node is used for acceleration.

node_comp_1 is a compute node using a Netronome SmartNIC in 4x10G Breakout mode. This node is **NOT** used for acceleration.

Note the lack of entry for node_comp_1 in env.ns_agilio_vrouter

File: /opt/contrail/utils/fabfile/testbeds/testbed.py

```
from fabric.api import env

#Management ip addresses of hosts in the cluster
node_ctrl_0 = 'root@172.18.47.35' # control node - no SmartNIC
node_comp_0 = 'root@172.18.47.39' # compute node - SmartNIC
node_comp_1 = 'root@172.18.47.37' # compute node - SmartNIC

#External routers if any
#for eg.
#ext_routers = [('mx1', '10.204.216.253')]
ext_routers = []

#Autonomous system number
router_asn = 64525

#Host from which the fab commands are triggered to install and provision
host_build = node_ctrl_0

#Role definition of the hosts.
env.roledefs = {
    'all': [node_ctrl_0, node_comp_0, node_comp_1],
    'cfgm': [node_ctrl_0],
    'openstack': [node_ctrl_0],
    'control': [node_ctrl_0],
    'compute': [node_comp_0, node_comp_1],
    'collector': [node_ctrl_0],
    'webui': [node_ctrl_0],
    'database': [node_ctrl_0],
    'build': [host_build],
    'storage-master': [node_ctrl_0],
    'storage-compute': [node_comp_0, node_comp_1],
}

# required if using env.ns_agilio_vrouter
control_data = {
    node_ctrl_0 : { 'ip': '172.18.48.32/24',
                    'gw': '172.18.48.1',
                    'device': 'eth1'},
    node_comp_0 : { 'ip': '172.18.48.33/24',
                    'gw': '172.18.48.1',
                    'device': 'nfp_p0'},
    node_comp_1 : { 'ip': '172.18.48.34/24',
```



```
        'gw': '172.18.48.1',
        'device': 'bond0'},
}

# Setup Netronome Agilio vRouter on specified nodes
# Populate to match compute node machine properties
env.ns_agilio_vrouter = {
    node_comp_0: {'huge_page_alloc': '24G',
                  'huge_page_size': '1G',
                  'coremask': '2,4',
                  'pinning_mode': 'auto:split'},
}

bond= {
    # 4x10G breakout mode on a 1x40G SmartNIC
    node_comp_1 : { 'name': 'bond0',
                    'member': ['nfp_p0', 'nfp_p1', 'nfp_p2', 'nfp_p3'],
                    'mode': '802.3ad',
                    'xmit_hash_policy': 'layer3+4' },
}

env.hostnames = {
    node_ctrl_0: 'node-ctrl-0',
    node_comp_0: 'node-comp-0',
    node_comp_1: 'node-comp-1',
}

#Openstack admin password
env.openstack_admin_password = 'netronome'

# Passwords of each host
env.passwords = {
    node_ctrl_0: 'netronome',
    node_comp_0: 'netronome',
    node_comp_1: 'netronome',
    host_build: 'netronome',
}

#For reimage purpose
env.ostypes = {
    node_ctrl_0: 'ubuntu',
    node_comp_0: 'ubuntu',
    node_comp_1: 'ubuntu',
}
```

5 Configuring OpenStack for Acceleration

5.1 Accelerating interfaces with virtio

Virtio instances must be configured to have their memory backed by hugepages. Configure the compute node for hugepages, and add the `hw:mem_page_size` extra_spec to all VM flavors that will be used with virtio accelerated instances.

To configure a VM flavor to use hugepage-backed memory:

```
nova flavor-key "$FLAVOR_UUID" set hw:mem_page_size=large
```

(This operation can also be accomplished using the Horizon GUI.)

Supported values for `hw:mem_page_size` with virtio acceleration are `large`, `2M`, and `1G`. If you configure a flavor with a specific page size (e.g., `1G`), VMs using the flavor will not be able to launch on a compute node that is configured with a different page size.

Use of `1G` hugepages is preferred for high performance network applications since they reduce TLB churn (although this effect is probably marginal in most cases). Note however that using `1G` hugepages may put a lower bound on the VM memory size, e.g., `2G`.

5.2 Accelerating interfaces with SR-IOV

All VM images are assumed to support virtio by default. To enable SR-IOV for an image, mark the with the `agilio.hw_acceleration_features` property. This property contains a comma-separated set of string tokens representing acceleration features.

- The token `SR-IOV` indicates that the image supports SR-IOV direct attach to accelerated hardware.
- Unknown tokens generate a warning.

To mark an image as compatible with SR-IOV hardware acceleration mode:

```
glance image-update "$IMAGE_UUID" --property agilio.hw_acceleration_features=SR-IOV
```

(This operation can also be accomplished using the Horizon GUI.)

6 Appendix

6.1 Contrail Links and Documents

This release of accelerated vRouter is targeted for Contrail version 3.1.

Contrail vRouter requirements and limitations are imposed on Netronome Agilio vRouter.

Please review the following links for Contrail documentation, code, and release notes:

- [Contrail Documentation / Feature Guide](#)
 - https://www.juniper.net/techpubs/en_US/contrail3.1/information-products/pathway-pages/contrail-feature-guide-pwp.html
- [Juniper GitHub](#)
 - <https://github.com/Juniper>
- [Contrail vRouter GitHub](#)
 - <https://github.com/Juniper/contrail-vrouter>
- [Contrail Release Notes](#)
 - http://www.juniper.net/techpubs/en_US/contrail3.1/information-products/topic-collections/release-notes/
- [Contrail Getting Started Guide](#)
 - https://www.juniper.net/techpubs/en_US/contrail3.1/information-products/pathway-pages/getting-started-pwp.html

6.2 Netronome Links and Documents

- [Netronome Website](#)
 - <https://www.netronome.com/>
- [Netronome GitHub](#)
 - <https://github.com/Netronome>
- [Accelerated vRouter Blueprint](#) - OpenContrail blueprint detailing the modifications required to OpenContrail to support a hardware offload API.
 - <https://github.com/Netronome/opencontrail-blueprints/blob/master/Hardware-Acceleration-Blueprint.md>

6.3 PCIe Quirk Kernel Patch

Linux kernels exhibit undesired behavior in PCIe configuration code. Netronome submitted a fix to the kernel maintainers for this issue which has been accepted into kernel version 4.5 and backported into the standard Ubuntu v3.13 and v4.4 kernels. If you are using an older kernel version, a patch needs to be applied to the kernel source code, or an already patched kernel needs to be installed. For your convenience, packages containing patched versions of the default kernels for various Ubuntu 14.04 LTS and RHEL/CentOS 7 variants are available on Netronome's support site (<http://support.netronome.com>).

Contact Netronome support if you require additional assistance.

Software for kernel versions can be found at <https://www.kernel.org/>. In the event Software cannot be retrieved, contact Netronome via support site (<http://support.netronome.com>) for additional assistance in obtaining.

This kernel patch code is free software; you can redistribute it and/or modify it under the terms of version 2 of the GNU General Public License as published by the Free Software Foundation.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details, a copy of this license has been provided in the accompanying README file.