# CONTENTS

1 Introduction 1

1.1 Key terms, acronyms and abbreviations .................................................. 1
1.2 Overview ........................................................................................................ 1

2 New features in plugin version 3.0.0 2

3 Limitations 3

4 Requirements 4

5 Installation Guide 5

5.1 Prerequisites ................................................................................................... 5
5.2 Installing Contrail Plugin .............................................................................. 5
5.3 Configuring Contrail Plugin .......................................................................... 6

6 DPDK-based vRouter 11

6.1 Description ................................................................................................. 11
6.2 Prerequisites ............................................................................................... 12
6.3 Restrictions .................................................................................................. 12
6.4 Configuration ............................................................................................... 13
6.5 Verification .................................................................................................. 15
6.6 DPDK related options .................................................................................. 16

7 SR-IOV 17

7.1 Prerequisites ............................................................................................... 17
7.2 Features ........................................................................................................ 17
7.3 What is SR-IOV ........................................................................................... 17
7.4 How to check if network interface is sriov capable, and how many VFs are available/enabled ................................................................. 17
7.5 How to enable SR-IOV in fuel ...................................................................... 18
7.6 How to create VM with sriov device ............................................................ 19

8 Using network templates 20

9 Verification 21

9.1 Basic checks ................................................................................................ 21
9.2 OSTF tests ................................................................................................... 22
9.3 Troubleshooting .......................................................................................... 24

10 Basic Contrail Operations 25

10.1 Logging in .................................................................................................... 26
10.2 Checking services status .................................................. 27
10.3 Creating the virtual networks .............................................. 27

11 Contrail Analytics Performance Tuning ................................. 29

12 Appendix ................................................................................. 30
CHAPTER ONE

INTRODUCTION

This document contains instructions for installing and configuring Contrail plugin for Fuel.

1.1 Key terms, acronyms and abbreviations

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Juniper Contrail</td>
<td>Contrail Cloud Platform is a foundational element of Juniper’s open cloud networking and NFV solutions.</td>
</tr>
<tr>
<td>SDN</td>
<td>Software defined network</td>
</tr>
<tr>
<td>RESTful API</td>
<td>Representational state transfer application programming interface</td>
</tr>
<tr>
<td>IDS</td>
<td>Intrusion detection system</td>
</tr>
<tr>
<td>DPI</td>
<td>Deep packet inspection</td>
</tr>
<tr>
<td>VIP</td>
<td>virtual IP address</td>
</tr>
<tr>
<td>BGP</td>
<td>Border gateway protocol</td>
</tr>
<tr>
<td>AS</td>
<td>Autonomous system</td>
</tr>
<tr>
<td>Contrail vRouter</td>
<td>Contrail vRouter is part of the compute node, which gets reachability information from the control plane and ensures native L3 services for host-based virtual machines.</td>
</tr>
<tr>
<td>MOS</td>
<td>Mirantis OpenStack</td>
</tr>
</tbody>
</table>

1.2 Overview

Contrail plugin for Fuel provides the functionality to add Contrail SDN for Mirantis OpenStack as networking backend option using Fuel Web UI in a user-friendly manner. Juniper Networks Contrail is an open software defined networking solution that automates and orchestrates the creation of highly scalable virtual networks.

Contrail features:

- Powerful API calls (REST or direct python class calls)
- Analytics engine: Traffic flow reports, statistics
- Network management at 2-4 OSI layers
- Service chaining architecture: you can transparently pass traffic through service instances, such as IDS, firewalls, DPI.
- Fine grained virtual network access policy control
NEW FEATURES IN PLUGIN VERSION 3.0.0

• Deployment is now role-based. Following roles are provided: Contrail-Control, Contrail-Config, Contrail-DB. This provides a possibility to deploy these components on different servers.

• VIPs for API and Web UI are now provided by Openstack Controllers and managed by Mirantis OpenStack HA. This provides a possibility to place Contrail components in different L2/L3 segments.

• Deployment tasks were rewritten to be more granular.

• DPDK-based vRouter. DPDK-based vRouter

• SR-IOV SR-IOV

• Plugin supports custom network templates feature of Fuel 7.0. Now it is possible to deploy a Contrail-enabled environment with reduced set of logical networks, e.g. Public, Management and Private nets can share the same interface. This simplifies routing configuration for large environments distributed across different L2 segments. More detailed information here Using network templates

• HTTPS on public endpoints. If HTTPS is enabled in Fuel UI, the same certificate will be used for Contrail API and Contrail WebUI.

• Contrail specific Ceilometer meters now supported.
• Removing Contrail-DB nodes from cluster is not supported by plugin, it can lead to data loss, so this must be a manual procedure. Adding new Contrail-DB nodes to the environment is supported.

• In case of using contrail service chaining with service instances, you may need to add neutron service user to a current tenant after you have deployed the environment:
  – Open Horizon dashboard, navigate to Identity - Projects page.
  – Click modify users button on the right side of admin project.
  – Add neutron user to project members with _member_ role.
The plugin has the following requirements for software and hardware:

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fuel version</td>
<td>7.0</td>
</tr>
<tr>
<td>Juniper Contrail version</td>
<td>3.0</td>
</tr>
<tr>
<td>Hardware</td>
<td>• At least 1 additional server for Contrail controller (contrail-control, contrail-config, contrail-db roles). Contrail-DB requires 320 Gb disk space.</td>
</tr>
<tr>
<td></td>
<td>• Additional network interface on each node except for MOS Controller.</td>
</tr>
</tbody>
</table>
CHAPTER FIVE

INSTALLATION GUIDE

5.1 Prerequisites

This guide assumes that you have installed Fuel and all the nodes of your future environment are discovered and functional.

5.2 Installing Contrail Plugin

1. Download Contrail plugin from the Fuel Plugins Catalog.
2. Copy the rpm downloaded at previous step to the Fuel Master node and install the plugin
   
   scp contrail-3.0-3.0.0-1.noarch.rpm <Fuel Master node ip>:/tmp/

3. Log into the Fuel Master node and install the plugin

   ssh <the Fuel Master node ip>
   fuel plugins --install contrail-3.0-3.0.0-1.noarch.rpm

   You should get the following output:
   
   Plugin <plugin-name-version>.rpm was successfully installed

4. Copy Juniper contrail install package (obtained from Juniper by subscription, more information can be found on official Juniper Contrail web-site ) to the Fuel Master node and run the installation script to unpack the vendor package and populate plugin repository

   scp contrail-install-packages_3.0.0.0-2723-kilo_all.deb \
   <Fuel Master node ip>:/var/www/nailgun/plugins/contrail-3.0/
   ssh <Fuel Master node ip> /var/www/nailgun/plugins/contrail-3.0/install.sh
5.3 Configuring Contrail Plugin

1. First, you need to create environment in Fuel UI.

2. Please select KVM or QEMU hypervisor type for your environment

3. Please select Neutron with tunneling segmentation network model. GRE segmentation is also supported, but you need to set it from Fuel CLI
4. If you plan to use Heat orchestration with autoscaling, you need to install Ceilometer too.

5. Activate the plugin and fill configuration fields with correct values:
   - AS number for BGP Gateway nodes communication: (defaults to 64512).
   - Gateway nodes IP addresses (provided as a comma-separated list) - peer addresses for BGP interaction with border routers.

6. Add nodes and assign them the following roles:
   - At least 1 Controller
   - At least 1 Compute
   - At least 1 node with Contrail-Control, Contrail-Config,Contrail-DB roles selected (3 or other odd number of nodes recommended for HA)
   - If you plan to use Heat with autoscaling, in addition to Ceilometer you need to add node with MongoDB role
These 3 roles are not necessary need to be on the same node. You can place them on different nodes if needed.

Sample node configuration is provided on a picture below.

7. Configure the disks on nodes with Contrail-DB role selected. The recommended size of partition for Contrail database is 256 GB or more.

8. Configure the network settings. See details at Mirantis OpenStack User Guide.
   - Open Nodes tab: Select all the nodes, push Configure interfaces button
• Set Private network to the separate network interface as the untagged network. **DO NOT USE THIS PHYSICAL INTERFACE FOR ANY OTHER NETWORK.** This interface will be used by contrail vRouter as the untagged port. It is recommended to set the bigger MTU for Private interfaces (e.g. 9000) if the switching hardware supports Jumbo Frames. This will enhance contrail network performance by avoiding packet fragmentation within Private network.

For other networking options please refer to Mirantis OpenStack User Guide. In case of using multiple L2 segments, you may need to configure networking according to the Operations Guide and supply static routes to BGP peers and other cluster networks in network_1.yaml file.

9. Example network configuration

Hardware servers with two network interfaces are used as openstack nodes. The interfaces configuration is following:

• Public, Management and Storage networks on the same interface with Admin net, using tagged VLANs

• The second interface is dedicated for Contrail operations as untagged (Private network)

**Warning:** Be sure to launch network verification check before starting deployment. Incorrect network configuration will result in non-functioning environment.

10. Press **Deploy changes** to deploy the environment.

After installation is finished, Contrail Web UI can be accessed by the same IP address as Horizon, but using HTTPS protocol and port 8143. For example, if you configured public network as described on screenshot below, then Contrail Web UI can be accessed by **https://10.109.1.3:8143**
Warning: First usable addresses from the Private network will be used as VIP for Contrail controllers. For example, if your Private network CIDR is 192.168.200.0/24, then Contrail VIP will be 192.168.200.1.
6.1 Description

The Data Plane Development Kit (DPDK) is a set of data plane libraries and network interface controller drivers for fast packet processing. The DPDK provides a programming framework for Intel x86 processors and enables faster development of high-speed data packet networking applications.

By default, contrail virtual router (vrouter) is running as a kernel module on Linux. The vrouter module is able to fill a 10G link with TCP traffic from a virtual machine (VM) on one server to a VM on another server without making any assumptions about hardware capabilities in the server NICs. Also, in order to support interoperability and use a standards-based approach, vrouter does not use new protocols/encapsulations. However, in network function virtualization (NFV) scenarios, other performance metrics such as packets-per-second (pps) and latency are as important as TCP bandwidth. With a kernel module, the pps number is limited by various factors such as the number of VM exits, memory copies and the overhead of processing interrupts.

In order to optimize performance for NFV use cases, vrouter can be integrated with the Intel DPDK (Data Plane Development Kit). To integrate with DPDK, the vrouter can now run in a user process instead of a kernel module.
This process links with the DPDK libraries and communicates with the vrouter host agent, which runs as a separate process. The application inside the guest VM can be written to use the DPDK API or it can use the traditional socket API. However, for NFV applications such as vMX, which require high performance, it would be preferable to use the DPDK API inside the VM.

6.2 Prerequisites

- Installed Fuel 7.0
- Installed contrail plugin Installation Guide
- Environment must be created with “KVM” for compute virtualization and “Neutron with tunneling segmentation” for networking
- Network card must support DPDK. List of compatible adapters can be found on DPDK website

6.3 Restrictions

- Only compute hosts can be configured with DPDK role. “DPDK role” is just a mark that enables DPDK feature on certain compute. If you try to use it with other roles it wouldn’t have any effect.
- Contrail DPDK feature doesn’t work with qemu virtualization as far as with nested KVM. This means that for current release DPDK-based vRouter works only on baremetal computes.
- Contrail DPDK vrouter permanently uses 1GB of hugepages, therefore, it is necessary to allocate enough amount of hugepages to run DPDK vrouter and VM’s(with DPDK) respectively.
6.4 Configuration

To enable DPDK you should proceed with following steps:

1. Enable contrail plugin in Fuel UI settings

2. Enable DPDK on Fuel UI

3. Choose the size and amount of huge pages to allocate. They will be used for both vRouter process
and VMs backing. 2MB sized huge pages can be added on-fly, 1GB sized require a reboot. Also, it is necessary to leave some amount of memory for the operating system itself.

4. Add DPDK role on computes where you want to have DPDK-based vRouter. **Computes that are not marked with DPDK role will use kernel-based vRouter.**

5. Deploy environment

**Warning:** Computes with DPDK-based vRouter require flavor with HugePages enabled. Instances with usual flavours can’t be launched on DPDK-enabled hosts.

If DPDK is enabled in plugin settings Fuel will create one flavor that will have hugepages support, named “m1.small.hpgs”. One can create custom flavor with following steps on controller node:

```
# . openrc
# nova flavor-create m2.small.hpgs auto 2000 20 2
# nova flavor-key m2.small.hpgs set hwmem_page_size=large
# nova flavor-key m2.small.hpgs set aggregate_instance_extra_specs:hpgs=true
```
6.5 Verification

After deploy finishes, you can verify your installation. First, proceed with basic checks.

1. Check that Contrail services and DPDK vrouter are running on compute node:

   ```
   root@node-37:~# contrail-status
   == Contrail vRouter ==
supervisor-vrouter: active
ccontrail-vrouter-agent active
ccontrail-vrouter-dpdk active
ccontrail-vrouter-nodemgr active
   ```

2. Check if DPDK vrouter catch interface:

   ```
   root@node-37:~# /opt/contrail/bin/dpdk_nic_bind.py -s
   Network devices using DPDK-compatible driver
   =============================================
   0000:06:00.0 '82599ES 10-Gigabit SFI/SFP+ Network Connection' drv=igb_uio unused=
   Network devices using kernel driver
   ====================================
   0000:02:00.0 '1350 Gigabit Network Connection' if=eth0 drv=igb unused=igb_uio
   0000:02:00.1 '1350 Gigabit Network Connection' if=eth1 drv=igb unused=igb_uio
   Other network devices
   ===============
   <none>
   ```

3. Check if vrouter use hugepages:

   ```
   root@node-37:~# grep Huge /proc/meminfo
   AnonHugePages: 0 kB
   HugePages_Total: 30000
   HugePages_Free: 29488
   HugePages_Rsvd: 0
   HugePages_Surp: 0
   Hugepagesize: 2048 kB
   ```

4. Check if vrouter utilize CPU:

   ```
   root@node-41:~# vif --list
   Vrouter Interface Table
   Flags: P=Policy, X=Cross Connect, S=Service Chain, Mr=Receive Mirror
   Mt=Transmit Mirror, T=Transmit Checksum Offload, L3=Layer 3, L2=Layer 2
   D=DHCP, Vp=Vhost Physical, Pr=Promiscuous, Vnt=Native Vlan Tagged
   Mnp=No MAC Proxy, Dpdk=DPDK PMD Interface, Rfl=Receive Filtering Offload
   Mon=Interface is Monitored, Uuf=Unknown Unicast Flood, Vof=VLAN insert/strip offload
   vif0/0 PCI: 0:0:0.0 (Speed 10000, Duplex 1)
   Type:Physical Nwaddr:00:1b:21:87:21:98 IPaddr:0
   RX device packets:3671 bytes:519373 errors:10
   RX port packets:3671 errors:0
   RX queue packets:5 errors:0
   RX queue errors to lcore 0 0 0 0 0 0 0 0 0 0 0 0
   RX packets:3671 bytes:499253 errors:0
   TX packets:4049 bytes:2135246 errors:0
   TX port packets:4049 errors:0
   TX device packets:4049 bytes:2135246 errors:0
   vif0/1 Virtual: vhost0
   Type:Host Nwaddr:00:1b:21:87:21:98 IPaddr:0
   Vrf0 Flags:LL2 MTU:1514 Ref:8
   RX device packets:45 errors:0
   RX queue errors to lcore 0 0 0 0 0 0 0 0 0 0 0 0
   RX packets:45 bytes:4322 errors:3
   TX packets:951 bytes:95940 errors:0
   TX queue packets:951 errors:0
   TX port packets:3771 errors:0
   vif0/2 Socket: unix
   Type:Agent Nwaddr:00:00:00:00:00:01:00 IPaddr:0
   Vrf:65535 Flags:L3 MTU:1514 Ref:2
   RX port packets:45 errors:0
   RX queue errors to lcore 0 0 0 0 0 0 0 0 0 0 0 0
   RX packets:45 bytes:4322 errors:3
   TX packets:951 bytes:95940 errors:0
   TX queue packets:951 errors:0
   ```

5. Check if vrouter create interface after creation VM:
6.6 DPDK related options

In this chapter described DPDK related options that you can change from Fuel UI:

- **“Enable DPDK feature for this environment.”** - this option enable DPDK globally, remember that you anyway must use “DPDK” role to mark compute where you want to have DPDK

- **“Hugepage size”** - Choose the size of huge pages that will be used for a dpdk feature. Check if 1GB pages are supported on the target compute node. 
  
  ```bash
  # grep pdpe1gb /proc/cpuinfo | uniq
  ```

- **“Hugepages amount (%))”** - set amount of memory allocated on each compute node for huge pages. It will use % of all memory available on compute. Remember that DPDK vrouter permanently use 1GB of huge pages and other applications running on compute node may not support huge pages, so this parameter should be used carefully.

- **“CPU pinning”** - this hexadecimal value describes how many and which exact processors will be used by dpdk-vrouter. CPU pinning is implemented using `taskset util`

- **“Patch Nova”** - current release (7.0) of MOS nova doesn’t have support for DPDK-based vRouter. In future, necessary patches will be included in MOS maintenance updates.

- **“Install Qemu and Libvirt from Contrail”** - DPDK-based vRouter needs huge pages memory-backing for guests. MOS 7.0 ships with qemu and libvirt that don’t support it. This is needed only for DPDK feature and will be implemented only on nodes where we have “DPDK” role.
7.1 Prerequisites

This guide assumes that you have installed Fuel and performed steps 5.3.1 - 5.3.9 from Installation Guide. To enable SR-IOV you need sriov capable network PCI card. Also, it is important to remember that only compute hosts can be configured with sriov role.

7.2 Features

1. You can have multiple VLANs inside one physical network

2. When using Passthrough (as in sriov scenario), there are no dhcp and metadata provided over openstack. You have to manage that manually or provide additional network port with usual openstack network.

7.3 What is SR-IOV

Quoting Mirantis blog post:

SR-IOV is a PCI Special Interest Group (PCI-SIG) specification for virtualizing network interfaces, representing each physical resource as a configurable entity (called a PF for Physical Function), and creating multiple virtual interfaces (VFs or Virtual Functions) with limited configurability on top of it, recruiting support for doing so from the system BIOS, and conventionally, also from the host OS or hypervisor. Among other benefits, SR-IOV makes it possible to run a very large number of network-traffic-handling VMs per compute without increasing the number of physical NICs/ports and provides means for pushing processing for this down into the hardware layer, off-loading the hypervisor and significantly improving both throughput and deterministic network performance.

7.4 How to check if network interface is sriov capable, and how many VFs are available/enabled

Issue following command on boostraped host:

```
lspci -s <bus ID> -vvv
```
7.5 How to enable SR-IOV in fuel

1. Enable SR-IOV in plugin settings and configure unique physnet name.

#. Assign SR-IOV role to compute hosts. SR-IOV will be enabled on all SR-IOV capable interfaces, not assigned to Fuel bridges(networks in Fuel UI).
1. Perform deploy as in 5.3.10 Installation Guide

7.6 How to create VM with sriov device

1. Create VN with configured physical network and vlan id:

   ```bash
   neutron net-create --provider:physical_network=<physical network from contrail settings tab> \
           --provider:segmentation_id=<Vlan_id> <Network_Name>
   ```

2. Create a subnet:

   ```bash
   neutron subnet-create <Network_name> <Subnet>
   ```

3. Create a Port:

   ```bash
   neutron port-create --fixed-ip subnet_id=<subnet uuid>,ip_address=<IP address from above subnet> \
            --name <name of port> <vn uuid> --binding:vnic_type direct
   ```

4. Boot VM with the port:

   ```bash
   nova boot --flavor m1.large --image <image name> --nic port-id=<uuid of above port> <vm name>
   ```
Starting from Fuel 7.0 it is possible to reduce the number of logical networks. This is implemented with the function called network templates. For detailed information on this feature, refer to Operations guide

This document provides sample configuration with network template. It is designed to get customers up and running quickly. The provided template utilizes 3 networks: Admin (PXE), Public and Private.

1. First do steps 5.3.1 - 5.3.7 from Installation Guide

2. Configure interfaces as shown on figure:

![Configure interfaces on 3 nodes](image)

3. Next, we need to set gateway for the private network. Here is how to do it with Fuel CLI:
   - Login with ssh to Fuel master node.
   - List existing network-groups
     ```
     fuel network-group --env 1
     ```
   - Remove and create again network-group private to set a gateway
     ```
     fuel network-group --delete --network 5
     fuel network-group --create --name private \\ --cidr 10.109.3.0/24 --gateway 10.109.3.1 --nodegroup 1
     ```
   - Set the `render_addr_mask` parameter to `internal` for this network by typing:
     ```
     fuel network-group --set --network 6 --meta '{"name": "private", "notation": "cidr", \ "render_type": null, "map_priority": 2, "configurable": true, "use_gateway": true, \ "render_addr_mask": "internal", "vlan_start": null, "cidr": "10.109.3.0/24"}'
     ```

4. Save sample network template

5. Upload the network template by typing:
   ```
   fuel --env 1 network-template --upload --dir /root/
   ```

6. Start deploy, pressing “Deploy changes” button.
After deploy finishes, you can verify your installation. First, proceed with basic checks described below.

### 9.1 Basic checks

1. Check that Contrail services are running.

   Login to Contrail controller node and run `contrail-status` command. All services should be in “active” state:

   ```bash
   # contrail-status
   == Contrail Control ==
   supervisor-control: active
   contrail-control: active
   contrail-control-nodemgr: active
   contrail-dns: active
   contrail-named: active
   == Contrail Analytics ==
   supervisor-analytics: active
   contrail-analytics-api: active
   contrail-analytics-nodemgr: active
   contrail-collector: active
   contrail-query-engine: active
   contrail-s Kemp-collector: active
   contrail-topology: active
   == Contrail Config ==
   supervisor-config: active
   contrail-api: active
   contrail-config-nodemgr: active
   contrail-device-manager: active
   contrail-discovery: active
   contrail-schema: active
   contrail-svc-monitor: active
   ifmap: active
   == Contrail Web UI ==
   supervisor-webui: active
   contrail-webui: active
   contrail-webui-middleware: active
   == Contrail Database ==
   supervisor-database: active
   contrail-database: active
   contrail-database-nodemgr: active
   kafka: active
   
   ```

2. Check the list of peers and peering status

   Login to Contrail WebUI, go to Monitor -> Control nodes, choose any and select a “Peers” tab. You should see your compute nodes(vRouters) and external router in a list of peers. Status should be “Established”
3. Check that external router was provisioned correctly:

   Login to Contrail WebUI, go to Configure -> Infrastructure -> BGP routers. Verify the IP address of router

   ![Contrail WebUI image]

   After that you can use health checks in Fuel UI, also called OSTF tests.

# 9.2 OSTF tests

- **Prerequisites for OSTF:**

  1. OSTF tests require two pre-defined networks created - net04 and net04_ext. The networks are created by Fuel during deployment. This section includes instructions how to create them if they were accidentally deleted. Floating IP addresses from net04_ext should be accessible from Fuel master node.

  2. 3 tests from “Functional tests” set require floating IP addresses. They should be configured on external router, routable from Fuel master node and populated in Contrail/Openstack environment.

  3. HA tests require at least 3 Openstack controllers.


- **OSTF networks and floating IPs configuration:**

  To create networks go to Contrail WebUI -> Configure -> Networking -> Networks

  1. Create network “net04”
2. Create network “net04_ext”.

It should be marked as “shared” and “external”

And have same route target as configured in external router

3. Allocate floating IP addresses from net04_ext

Go to Contrail WebUI -> Configure -> Networking -> Manage Floating IPs
After checking networks and floating IP addresses, start OSTF tests. For more details, refer to Fuel user-guide.

### 9.3 Troubleshooting

Start with checking output of contrail-status command. Then check the logs for corresponding service. Contrail logs are located in `/var/log/contrail/` directory, and log names match with contrail service name. Cassandra logs are located in `/var/log/cassandra/`, zookeeper’s is in `/var/log/zookeeper/`. 
BASIC CONTRAIL OPERATIONS

This document describes very basic operations with Contrail UI. For detailed information on Contrail operations, please refer to official Juniper documentation.
10.1 Logging in

Log into Contrail UI using the OpenStack admin credentials.
10.2 Checking services status

Verify the status of Contrail Control Analytics and Config nodes along with vRouters in *Infrastructure* using the *Dashboard* tab of the left-hand *Monitor* menu.

10.3 Creating the virtual networks

- Open left-hand *Configure* menu and click *Networking* option. Enter *Networks* tab and use “+” sign at the right side to create a new virtual network. Enter the network name and add an IP subnet. Gateway address will be added automatically.

- To create an external network, you need to add *Shared* and *External* flags to the created network using the *Advanced Options* sections and provide a proper Routing mark in Route Targets section to let this network to be announced to the public routing table. The Routing mark is two numbers divided by a semicolon, e.g. 64512:10000.
10.3. Creating the virtual networks
CONTRAIL ANALYTICS PERFORMANCE TUNING

For large installations, the following changes are required to achieve better performance of contrail database.

By default, the analytics_ttl values are set to -1. We recommend changing the default number to ensure the best performance in highly scaled out environments. An example configuration is as follows:

1. /etc/cassandra/cassandra-env.sh
   ```
   JVM_OPTS="JVM_OPTS -XX:MaxTenuringThreshold=1"
   JVM_OPTS="JVM_OPTS -XX:MaxTenuringThreshold=30"
   ```

2. /etc/contrail/contrail-collector.conf
   ```
   analytics_config_audit_ttl=2160
   analytics_statistics_ttl=24
   analytics_flow_ttl=2
   ```

Note: Please restart the supervisor services (supervisor-database, supervisor-analytics) after making these changes.
1. Contrail overview.
3. Contrail major components.
4. Contrail architecture.
5. Contrail quick start guide.