

# Contrail Release 4.0.1 Release Notes

Release 4.0.1  
September 2017

## Contents

Introduction .....	3
New and Changed Features .....	3
New and Changed Features in Contrail Release 4.0.1 .....	3
Contrail EVPN-VXLAN Support Using QFX Series Switches .....	3
Installation of Contrail Release 4.0.1 .....	4
Provisioning Contrail CNI for Kubernetes .....	4
Using Helm to Provision Contrail with Kubernetes .....	4
New and Changed Features in Contrail Release 4.0 .....	4
Alarms History .....	4
Allowed Address Pair .....	4
Containerized Contrail .....	4
Contrail Integration with Kubernetes .....	5
Mapping VLAN Tags from Physical NIC to a VMI (NIC-Assisted mirroring) .....	5
TCP Segment Offload (TSO) for DPDK vRouter .....	6
Platform Support .....	6
OpenStack Application Support .....	6
Supported Platforms Contrail 4.0.x .....	6
Known Behavior .....	7
Known Behavior in Contrail Release 4.0.1 .....	7
Known Behavior in Contrail Release 4.0 .....	8
Resolved Issues .....	11
Resolved Issues in Contrail Release 4.0.1 .....	11
Resolved Issues in Contrail Release 4.0 .....	11
Deprecated Items .....	11
Contrail Discovery Service and IF-MAP .....	11
Configuration with Testbed.py .....	12
Fabric Commands .....	12
Upgrading Contrail 3.2 to 4.0 .....	12
Overview of Changes in Contrail Release 4.0 .....	12
Assumptions .....	13
Using a Server Manager Script to Upgrade from Contrail 3.2 to 4.0 .....	13
Running the Upgrade Script .....	13

Steps for Upgrading Contrail 3.2 to 4.0 (Without using Server-Manager for upgrade) . . . . .	13
Installing Red Hat OpenShift Container Platform with Contrail Networking . . . . .	15
Installing Red Hat OpenShift Container Platform . . . . .	16
Installing Contrail Networking for Red Hat OpenShift . . . . .	17
Initializing Red Hat OpenShift Container and Contrail Networking . . . . .	17
Requesting Technical Support . . . . .	19
Self-Help Online Tools and Resources . . . . .	20
Opening a Case with JTAC . . . . .	20
Revision History . . . . .	20

---

## Introduction

---

Juniper Networks Contrail is an open, standards-based software solution that delivers network virtualization and service automation for federated cloud networks. It provides self-service provisioning, improves network troubleshooting and diagnostics, and enables service chaining for dynamic application environments across enterprise virtual private cloud (VPC), managed Infrastructure as a Service (IaaS), and Networks Functions Virtualization (NFV) use cases.

These release notes accompany Release 4.0.1 of Juniper Networks Contrail. They describe new features, limitations, and known problems.

These release notes are displayed on the Juniper Networks Contrail Documentation Web page at [http://www.juniper.net/techpubs/en\\_US/contrail4.0/information-products/topic-collections/release-notes/index.html](http://www.juniper.net/techpubs/en_US/contrail4.0/information-products/topic-collections/release-notes/index.html).

---

## New and Changed Features

---

The features listed in this section are new or changed as of the listed release. A brief description of each new or changed feature is included.

- [New and Changed Features in Contrail Release 4.0.1 on page 3](#)
- [New and Changed Features in Contrail Release 4.0 on page 4](#)

### New and Changed Features in Contrail Release 4.0.1

The feature listed in this section is new as of Contrail Release 4.0.1.

---

#### Contrail EVPN-VXLAN Support Using QFX Series Switches

---

Contrail Release 4.0.1 enables you to use Ethernet VPN (EVPN) with Virtual Extensible LAN protocol (VXLAN) encapsulation when you have an environment that includes both virtual and bare-metal devices. MX Series routers use EVPN-VXLAN encapsulation to provide both Layer 2 and Layer 3 connectivity for end stations within a Contrail virtual network (VN).

Two types of encapsulation methods are used in virtual networks:

- MPLS-over-GRE (generic routing encapsulation) is used for Layer 3 overlay virtual network routing between Contrail and MX Series routers.
- EVPN-VXLAN is used for Layer 2 overlay virtual network connectivity between virtual machines on Contrail, bare-metal servers attached to QFX Series switches, and their respective Layer 3 gateway configured on the QFX Series switch. Subsequently, inter-VXLAN routing between virtual machines and bare-metal servers, and between bare-metal servers on different VXLAN network identifiers (VNIs), is performed on the QFX Series switch.

For more information, see

[https://github.com/Juniper/contrail-controller/blob/master/specs/qfx\\_evpn.md](https://github.com/Juniper/contrail-controller/blob/master/specs/qfx_evpn.md).

### Installation of Contrail Release 4.0.1

---

Starting with Contrail Release 4.0.1, Contrail installation has no outside dependency and is fully offline. Previous versions of Contrail installation had dependencies on open source packages available through internet repositories. Because the location of internet repositories can change, starting with Contrail Release 4.0.1, the installation has no outside dependencies. Previous versions of Contrail might also require manual installation of the Cobbler package.

### Provisioning Contrail CNI for Kubernetes

---

You can provision a Contrail Container Network Interface (CNI) for Kubernetes in Contrail Release 4.0.1. For more information, see [Provisioning Contrail CNI for Kubernetes](#).

### Using Helm to Provision Contrail with Kubernetes

---

Helm is a tool that helps package, install, and manage Kubernetes applications. Starting with Contrail Release 4.0.1, you can use Helm to provision Contrail with Kubernetes. For more information, see [Using Helm to Provision Contrail with Kubernetes](#).

## New and Changed Features in Contrail Release 4.0

The features listed in this section are new as of Contrail Release 4.0.

### Alarms History

---

New fields in the Contrail Web user interface at **Monitor > Alarms > Dashboard** display alarms history, including alarms that were set or reset. You can also use a **contrail-status** query to view the alarms history. The **contrail-status** also displays a history of added, updated, and removed information for UVEs in Contrail.

See [Alarms History](#).

### Allowed Address Pair

---

An allowed address pair extension is an OpenStack feature supported by Contrail. By default, there is no way to specify additional MAC and IP address pairs that are allowed to pass through a port in Neutron, because ports are locked down to their MAC address and the fixed IPs associated with their port for anti-spoofing reasons. This locking can sometimes prevent protocols such as VRRP from providing a high availability failover strategy. Using the allowed address pair extension enables additional MAC and IP address pairs to be allowed through ports in Neutron.

For more information, see [Service Chain Version 2 with Port Tuple](#).

### Containerized Contrail

---

Starting with Contrail Release 4.0, some of the Contrail subsystems are delivered as Docker containers that group together related functional components.

The default containerized components include:

- contrail-controller
- contrail-analytics
- contrail-analyticsdb
- contrail-lb (loadbalancer)

Key features of the new architecture of Contrail containers:

- All of the containers are multiprocess Docker containers.
- Each container has an INI-based file that holds the configurations for all of the applications running in that container.
- Each container is self-contained, with minimal external orchestration needs.
- The user toolset `contrailctl` is used to manage the container configuration files.

To install containerized Contrail, all users need to use Server Manager or Server Manager Lite, even if they will not be using Server Manager for other provisioning functions.

For more information, see:

- [Introduction to Containerized Contrail Modules](#)
- [Installing Containerized Contrail Clusters Using Server Manager](#)
- [Installing Containerized Contrail Using Server Manager Lite \(SM-Lite\)](#)
- [Configuring Services Within Contrail Containers](#)
- [High Availability for Containerized Contrail](#)

---

### Contrail Integration with Kubernetes

Contrail Release 4.0 supports a Container Network Interface (CNI) for integrating Contrail with the Kubernetes automation platform. Kubernetes, also called K8s, is an open-source platform for automating deployment, scaling, and operations of application containers across clusters of hosts, providing container-centric infrastructure. It provides a portable platform across public and private clouds. Kubernetes supports a pluggable framework called Container Network Interface (CNI) for most of the basic network connectivity. Contrail Release 4.0 provides support for CNI for Kubernetes.

For more information, see [Contrail Integration with Kubernetes](#).

---

### Mapping VLAN Tags from Physical NIC to a VMI (NIC-Assisted mirroring)

Contrail Release 4.0 has the ability to mirror specific traffic to a traffic analyzer or to a physical probe using the Network interface card (NIC) instead of the vRouter to mirror packets. When NIC-assisted mirroring is enabled, ingress packets to be mirrored sent from a VM are routed to the NIC with a configured VLAN tag. The NIC is configured for VLAN port-mirroring and mirrors any packet with the VLAN tag.

See [Mapping VLAN Tags from a Physical NIC to a VMI \(NIC-Assisted Mirroring\)](#).

### TCP Segment Offload (TSO) for DPDK vRouter

The ability to provide TCP segment offload (TSO) has been added to DPDK v routers. The TSO feature is already available on non-DPDK routers. The TSO feature helps provide increased performance support for TCP-based large data applications, such as Web servers. The TSO feature enables segmentation and reassembly of large TCP segments to occur in lower layers of the stack, thereby boosting performance.

### Platform Support

Contrail Release 4.0 includes the following platform support:

- Kubernetes 1.6 on Ubuntu 16.04.2
- Kubernetes 1.6 on CentOS 7.2
- Openstack Mitaka on Ubuntu 14.04.5
- Openstack Newton on Ubuntu 16.04.2

### OpenStack Application Support

Contrail Release 4.0 supports integration with Fuel. Fuel is an open source deployment and management tool for OpenStack to streamline and accelerate the process of deploying, testing, and maintaining various configurations of OpenStack at scale.

## Supported Platforms Contrail 4.0.x

Table 1 on page 6 lists the operating system versions and the corresponding Linux or Ubuntu kernel versions supported by Contrail Release 4.0.

**Table 1: Supported Platforms**

Contrail Release	OpenStack Release	Operating System and Kernel Versions
Contrail Release 4.0.1	OpenStack Ocata	<ul style="list-style-type: none"> <li>• Ubuntu 16.04.2—Linux kernel version 4.4.0-62-generic</li> <li>• VMware vCenter 6.0, 6.5—Ubuntu 16.04.2 kernel version 4.4.0-62-generic</li> </ul>
	OpenStack Newton	<ul style="list-style-type: none"> <li>• Ubuntu 16.04.2—Linux kernel version 4.4.0-62-generic</li> <li>• VMware vCenter 6.0, 6.5—Ubuntu 16.04.2 kernel version 4.4.0-62-generic</li> </ul>
	OpenStack Mitaka	<ul style="list-style-type: none"> <li>• Ubuntu 14.04.5—Linux kernel versions 3.13.0-110-generic and 4.4.0-34-generic</li> <li>• VMware vCenter 6.0, 6.5—Ubuntu 14.04.4 kernel version 3.13.0-110-generic</li> </ul>

Table 1: Supported Platforms (*continued*)

Contrail Release	OpenStack Release	Operating System and Kernel Versions
Contrail Release 4.0	OpenStack Newton	<ul style="list-style-type: none"> <li>• Ubuntu 16.04.2—Linux kernel version 4.4-62</li> <li>• Red Hat 7.3—Linux kernel version 3.10.0-514.6.2</li> <li>• VMware vCenter 5.5, 6.0—Ubuntu 14.04.4 kernel version 3.13.0-106-generic</li> </ul>
	OpenStack Mitaka	<ul style="list-style-type: none"> <li>• Ubuntu 14.04.5—Linux kernel versions 3.13.0-106-generic and 4.4.0-34-generic</li> <li>• VMware vCenter 5.5, 6.0—Ubuntu 16.04.2 kernel version 4.4.0-62-generic</li> </ul>



**NOTE:** In Contrail Release 4.0 and later, if the stock kernel version of your Ubuntu system is older than the required version, you can upgrade the kernel for all nodes in the cluster by using the following parameter in `cluster.json` for Server Manager or SM-Lite provisioning or `testbed.py`.

```
{
  "cluster" : [{
    "parameters" : {
      "provisioning" : {
        "contrail" : {
          "kernel_upgrade" : true
        }
      }
    }
  }]
}
```

## Known Behavior

This section lists known limitations with this release. Bug numbers are listed and can be researched in [Launchpad.net](https://bugs.launchpad.net/juniperopenstack) at <https://bugs.launchpad.net/juniperopenstack>.

- [Known Behavior in Contrail Release 4.0.1 on page 7](#)
- [Known Behavior in Contrail Release 4.0 on page 8](#)

### Known Behavior in Contrail Release 4.0.1

- 1681680 When the DPDK vRouter fragments packets before sending them on the wire, the reassembly of the fragments on the receiver might time out in some cases.
- 1686236 Provisioning Keystone v3 support in RHOSP10 deployment with Contrail is not implemented.
- 1690904 TLS configured LBaaS needs Barbican service on OpenStack node and Barbican client on compute nodes. As a workaround, perform the following steps:
  1. Install the Barbican service on OpenStack node.
  2. Install the Barbican client on all compute nodes.

3. Configure the `/etc/contrail/contrail-lbaas-auth.conf` file with the required authentication information on all compute nodes. For example, following is a sample content of the `/etc/contrail/contrail-lbaas-auth.conf` file:

```
[BARBICAN]
admin_tenant_name=service
admin_user=neutron
admin_password=<neutron-service-passwd>
auth_url=http://<keystone-service-ip>:<keystone-service-port>/v2.0
region=RegionOne
```

- 1693590 Cinder volume creation fails after provisioning. As a workaround, restart the cinder-volume service manually on all OpenStack nodes.
- 1694572 Only one of the interfaces listed under a server can have a default gateway.
- 1698387 Contrail automatically sets a password during SM-Lite provisioning using the example JSONs in [Sample JSONs](#). To view the password, use the following command:  
**server-manager display cluster --cluster\_id cluster1 -d -s | grep admin\_password**
- 1705795 On a RHOSP10 provisioned cluster, if the vrouter-agent gets restarted, vhost0 interface does not come up. The **service supervisor-vrouter restart** command brings the service back up.
- 1711256 Project isolation is not supported in nested mode. In nested mode, Namespaces-isolation results in a virtual-network creation. It doesn't create a new project.

## Known Behavior in Contrail Release 4.0

- 1623695 In case of RBAC enabled clusters, user should create network-ipam in their own tenant configuration instead of using the default network-ipam for which the user doesn't have permissions.
- 1624148 In case of RBAC enabled clusters, service instance automatically created by the system on behalf of a user will not be visible in the UI.
- 1650420 In case of RBAC enabled clusters, objects created through LBaaS plugin are created with Neutron ownership.
- 1675224 VMs in non-HA cluster in SHUTOFF are in shutdown state after upgrade. As a workaround, set **resume\_guests\_state\_on\_host\_boot = True** in the `nova.conf` file of the compute node, for the guest VMs to be resumed.
- 1681680 When the DPDK vRouter fragments packets before sending them on the wire, the reassembly of the fragments on the receiver might time out in some cases.
- 1689761 SSL communication to Keystone service cannot be configured using Server Manager
- 1690108 In case of Contrail HA deployment, the lb container must not be installed on same host with other contrail containers.
- 1690904 TLS configured LBaaS needs Barbican service on OpenStack node and Barbican client on compute nodes. As a workaround, perform the following steps:
  1. Install the Barbican service on OpenStack node.



2. Install the Barbican client on all compute nodes.
3. Configure the `/etc/contrail/contrail-lbaas-auth.conf` file with the required authentication information on all compute nodes. For example, following is a sample content of the `/etc/contrail/contrail-lbaas-auth.conf` file:

```
[BARBICAN]
admin_tenant_name=service
admin_user=neutron
admin_password=<neutron-service-passwd>
auth_url=http://<keystone-service-ip>:<keystone-service-port>/v2.0
region=RegionOne
```

- 1691245 Kernel crashes when enabling LBaaS on Linux kernel version 4.4 and DPDK Hypervisor.
- 1691862 Server Manager requires internet access for installation of certain packages such as Cobbler, Puppet, DHCP, bind, and tftp.
- 1692067 Requests to Contrail Web-UI are not load balanced through contrail-lb. To access Web-UI, you must connect using the system IP on which the controller Docker runs.
- 1693590 Cinder volume creation fails after provisioning. As a workaround, restart the cinder-volume service manually on all OpenStack nodes.
- 1694343 In DPDK vRouter use-cases (SNAT, LBaaS) that require netns to be launched, do not set Jumbo frames. Use MTU <= 1500 bytes.
- 1694519 On a 16.04.2 Newton cluster, even in case of non-DPDK vRouter, the "contrail-status" will show status for "contrail-vrouter-dpdk" and during a restart of vRouter service, a core might appear. This does not have any functionality impact. As a workaround, to avoid seeing the core, remove the `/lib/systemd/system/contrail-vrouter-dpdk.service` file from the non-DPDK compute nodes.
- 1694849 On Ubuntu 16.04-based Docker containers sometimes the ProcessStatus alarm is not activated even if a process is not running. As a workaround, restart the node manager process based on the process that is not running on the respective container. For example, for the controller Docker, either the `contrail-control-nodemgr` or `contrail-config-nodemgr` process must be restarted. On the analytics Docker, `contrail-analytics-nodemgr` process needs to be restarted.
- 1694851 Provisioning of non-HA cluster using Server-Manager UI does not work in Contrail Release 4.0. However, CLI and JSON-based provisioning work for all use-cases.
- 1694857 After upgrade of cluster running Contrail Release 3.2 to 4.0, sometimes the Neutron server process does not start automatically. As a workaround, start the Neutron server explicitly after upgrade.
- 1695544 LBaaS cannot be configured from Horizon UI. Use CLIs instead of UI.
- 1695559, 1695503 Usage of uppercase characters in variable values in JSONs might stall installation. Some of the known variables where upper case characters cannot be used are server host names and image id.

- 1695566 CEPH storage integration with Ubuntu 16.04.2 Newton is not supported. Integration with Ubuntu 14.04.5 Mitaka is supported.
- 1695584 Server Manager provisioning-support for SSL communication for Keystone service is not available.
- 1695662 In Ubuntu 16.04.2 Newton cluster, the rabbitmq issue described in [Rabbitmq boot failure with "tables\\_not\\_present"](#) might be seen. As a workaround, perform the following steps:
  1. Stop the rabbitmq service on the node which is not clustered .

```
service stop and epmd -kill
```

In a three node setup, mostly two of nodes are clustered together and the third one is single. It's also possible that all the rabbitmq nodes are single too.
  2. Remove the `/var/lib/rabbitmq/mnesia` directory.
  3. Re-start rabbitmq on the node.
  4. Check Nova service-list output and restart all the services which are marked down.
- 1695741, 1694368 Ceilometer provisioning using Server-Manager will not work. If the user enables provisioning knob to enable ceilometer, sometimes provisioning will not get completed.
- 1695770 SSL communication to API-server provisioned using Server Manager will not work.
- 1695792 For use-cases where VMs on a compute node bypass vRouter through SRIOV, the nova.conf should have options to exclude interface used by vVouter. The Contrail Release 4.0 Ansible-based provisioning code does not add this. You need to explicitly set it as follows:
  1. Edit nova.conf in all OpenStack nodes and add the following lines.

```
scheduler_available_filters = nova.scheduler.filters.all_filters  
scheduler_default_filters = RetryFilter, AvailabilityZoneFilter, RamFilter,  
ComputeFilter, ComputeCapabilitiesFilter, ImagePropertiesFilter,  
PciPassthroughFilter
```
  2. Restart Nova scheduler service.
- 1695842 Server Manager provisioning support for vcenter-only and vcenter-as-compute use-cases is not available. Use fab commands to install vcenter-only and vcenter-as-compute use-cases.
- 1695845 In Contrail Release 4.0, on a DPDK compute, the VMs cannot bypass vrouter through SRIOV. This use-case works in non-DPDK computes.
- 1695856 Server Manager Ansible-based provisioning support for Keystone V3 for containerized deployment is not available.
- 1695859 Server Manager does not have Ansible-based provisioning support for secure communication for rabbitmq service.
- 1695938 For use-cases that requires VMs in computes bypass vRouter using SRIOV, nova.conf should be configured with a white list of VFs that Nova can use for attaching

VMs. The Contrail Release 4.0 Ansible-based provisioning code does not add this. You need to explicitly update the nova.conf similar to the following example and restart the nova-compute service on the compute nodes:

```
pci_passthrough_whitelist = { "address": "0000:81:10.0", "physical_network":
    "physnet1" }
pci_passthrough_whitelist = { "address": "0000:81:10.2", "physical_network":
    "physnet1" }
pci_passthrough_whitelist = { "address": "0000:81:10.4", "physical_network":
    "physnet1" }
pci_passthrough_whitelist = { "address": "0000:81:10.6", "physical_network":
    "physnet1" }
pci_passthrough_whitelist = { "address": "0000:81:11.0", "physical_network":
    "physnet1" }
pci_passthrough_whitelist = { "address": "0000:81:11.2", "physical_network":
    "physnet1" }
pci_passthrough_whitelist={"devname":"p1p2", "physical_network":"physnet2"}
```

## Resolved Issues

This section lists limitations that are resolved with this release.

- [Resolved Issues in Contrail Release 4.0.1 on page 11](#)
- [Resolved Issues in Contrail Release 4.0 on page 11](#)

### Resolved Issues in Contrail Release 4.0.1

You can research limitations that are resolved with this release in Launchpad at:

<https://launchpad.net/juniperopenstack/r4.0/r4.0.1.0>

### Resolved Issues in Contrail Release 4.0

You can research limitations that are resolved with this release in Launchpad at:

<https://launchpad.net/juniperopenstack/+milestone/r4.0.0.0-fcs>

## Deprecated Items

The following features have been deprecated in Contrail Release 4.0.

### Contrail Discovery Service and IF-MAP

Starting with Contrail Release 4.0, the existing centralized Contrail discovery service is replaced with a distributed method of allocating service resources. Any discovery APIs will cease to work. The replacement is a distributed resource allocation list of service nodes, maintained in each module of the system.

Additionally, information previously managed by IF-MAP has been changed to items managed in CONFIGDB.

For more information, see [Distributed Service Resource Allocation with Containerized Contrail](#).

## Configuration with Testbed.py

In previous versions of Contrail, a **testbed.py** was used to initiate configurations at installation. With Contrail Release 4.0, the use of **testbed.py** is limited to only those installations that are managed by SM-Lite. Most users will edit JSON files for the initial configuration at installation. Installation using JSON files can be performed with Server Manager or SM-Lite.

## Fabric Commands

Contrail Release 4.0 has limited support for Fabric commands. Installations that were previously accomplished with Fabric commands are now accomplished by means of Server Manager or SM-Lite deployments.

## Upgrading Contrail 3.2 to 4.0

---

Contrail Release 4.0 presents a number of differences from earlier versions of Contrail, the most notable is that many Contrail services are now run within containers.

This section provides the process for upgrading an existing Contrail Release 3.2 system to Contrail Release 4.0. You can perform an upgrade by using a convenient script, or by performing the upgrade manually. Both procedures are included in this topic.

- [Overview of Changes in Contrail Release 4.0 on page 12](#)
- [Using a Server Manager Script to Upgrade from Contrail 3.2 to 4.0 on page 13](#)
- [Steps for Upgrading Contrail 3.2 to 4.0 \(Without using Server-Manager for upgrade\) on page 13](#)

## Overview of Changes in Contrail Release 4.0

For previous releases of Contrail, through Contrail Release 3.2.x, Contrail could be provisioned by using fab commands or by using the Contrail Server Manager. Starting with Contrail Release 4.0, many Contrail services have been containerized, and provisioning can only be accomplished by using the Contrail Server Manager.

Additionally, the use of fab commands is no longer supported in Contrail Release 4.0.

You can perform the upgrade by using the steps presented here, or by using a script available in Server Manager that automates the steps.

Significant differences between Contrail 3.2 and 4.0 include:

- A number of Contrail services are run in containers. Container default names in Contrail Release 4.0 include:
  - contrail-controller
  - contrail-analytics
  - contrail-analyticsdb
  - contrail-lb (load-balancer)

For more information about Contrail containers, see *Introduction to Containerized Contrail Modules*.

### Assumptions

The upgrade procedure assumes that RabbitMQ server and Neutron server are running on the OpenStack node by default.

## Using a Server Manager Script to Upgrade from Contrail 3.2 to 4.0

A script is available in Server Manager that runs the steps provided in *Steps for Upgrading Contrail 3.2 to 4.0*.

The script location is: `/opt/contrail/server_manager/inplace_upgrade.py`.

Information regarding using the script:

- The script uses the **openstack-config** utility, you can install the **contrail-setup** package to get the utility.
- The timeout for cluster provisioning completion is 3600 seconds, you can increase the timeout for a cluster with a large number of nodes.

The default timeout parameter in the script:

**PROV\_TIMEOUT = 3600**

- The script execution log is saved at `/var/log/contrail/inplace_upgrade.log`.

### Running the Upgrade Script

1. Add the image, cluster, and servers to Server Manager running Contrail Release 4.0 code.
2. Run the script.

```
user@node:/opt/contrail/server_manager# python inplace_upgrade.py -h
usage: inplace_upgrade.py [-h] cluster_name image_name
```

positional arguments:

```
cluster_name  cluster to be upgraded. The cluster and servers should be
              existing in SM
image_name    version to upgrade to
```

optional arguments:

```
-h, --help    show this help message and exit
```

## Steps for Upgrading Contrail 3.2 to 4.0 (Without using Server-Manager for upgrade)

This procedure lists the steps needed for upgrading Contrail 3.2 to 4.0.

This procedure is an in-service upgrade from a running 3.2 system that will temporarily run in parallel with the new 4.0 system.

The services and roles referred to in this procedure could be running in separate nodes. Run the procedure commands in the nodes that are running the relevant roles or services.

1. Stop the listed services.

- supervisor-analytics
- supervisor-support-service
- supervisor-database
- contrail-database
- supervisor-webui
- supervisor-config
- supervisor-control
- haproxy (all nodes)
- redis-server (analytics/webui node)
- memcached (config nodes)
- neutron-server (config nodes)
- zookeeper (config nodes)

Example stop commands include:

**service supervisor-analytics stop**

**service supervisor-support-service stop**

2. Stop the epmd process. The epmd process is a RabbitMQ service that is usually running on a config node.

3. Use a **status** command to verify that none of the services that were stopped in Step 1 are still running.

Example status commands include:

**service supervisor-analytics status**

**service supervisor-support-service stop status**

4. Prepare the cluster.json and server.json, as required for Server Manager provisioning for Contrail Release 4.0, see *Installing Containerized Contrail Clusters Using Server Manager*.

5. Provision the new 4.0 cluster and wait for provisioning to be completed.

6. Check the status of provisioning.

**server-manager status server --cluster\_id <cluster-id>**

7. To prevent port conflict with Cassandra between the two versions of Contrail, on the 3.2 config node(s), edit the Cassandra configuration file at `/etc/cassandra/cassandra.yaml` to change the listening port from 9160 to 29160.

When finished, start the Cassandra service.

```
rpc_port: 29160
```

```
service cassandra start
```

8. Identify the new (4.0) and old (3.2) IP address lists for Cassandra.
  - a. Connect to the controller container.
  - b. Edit `/usr/lib/python2.7/dist-packages/contrail_issu/issu_contrail_config.py` and add the old Cassandra list and the new Cassandra list:

```
'old_cassandra_address_list': '192.xxx.xxx.102:29160',
```

```
'new_cassandra_address_list': '10.xx.5.xxx:9161',
```

It is sufficient to provide a single ip:port of the Contrail 3.2 Cassandra node.

9. Run a Cassandra sync between the old and new config nodes, to copy the cassandra config keyspaces from the release 3.2 nodes to the Contrail 4.0 Cassandra database.

```
contrail-issu-pre-sync
```

Wait for the command to complete.

10. Shut down the Cassandra service in the 3.2 controller.
11. The upgrade of the compute nodes includes a reboot of the compute nodes. The reboot turns off the guest virtual machine instances. To restart the virtual machine instances, log in to the Openstack node and restart the VMs:

```
nova start <instance-uuid>
```

---

## Installing Red Hat OpenShift Container Platform with Contrail Networking

Perform the following steps to install Red Hat OpenShift Container Platform with Juniper Networks Contrail Networking.

- [Installing Red Hat OpenShift Container Platform on page 16](#)
- [Installing Contrail Networking for Red Hat OpenShift on page 17](#)
- [Initializing Red Hat OpenShift Container and Contrail Networking on page 17](#)

## Installing Red Hat OpenShift Container Platform

To install Red Hat OpenShift Container Platform.

1. Install CentOS 7.3 minimal distribution on both master and slave nodes.
2. Download the Contrail-Ansible package and the Contrail-Kubernetes-Docker-Images package from Juniper's download site and copy them to the master node. You must create an account to download the packages.
3. Install the Extra Packages for Enterprise Linux (EPEL) on the master and slave nodes.

```
(all-nodes)# yum install git epel-release vim NetworkManager -y
(all-nodes)# yum update -y
```

4. Install all dependencies, set SELinux to "enforcing", and reboot the nodes.

```
(master)# yum install kernel-devel kernel-headers -y
(master)# yum install ansible pyOpenSSL python-cryptography python-lxml -y
(slave)# yum install kernel-devel kernel-headers nfs-utils socat -y
(all-nodes)# vi /etc/sysconfig/selinux
SELINUX=enforcing
(all-nodes)# reboot
```

5. Enable password-free SSH access to all nodes from the master node and verify that you can log in without a password.

```
(master)# ssh-keygen -t rsa
(master)# ssh-copy-id <root@<master-node-ip>
(master)# ssh-copy-id <root@<slave-node-ip>
```

6. Add hostnames of all nodes to the `/etc/hosts` file and verify that you can resolve the hostnames to IP addresses.

```
(all-nodes)# cat /etc/hosts
192.0.2.0 5b4s40.device.example.net 5b4s40 #master-node
192.0.2.1 5b4s41.device.example.net 5b4s41 #slave-node
(master)# ping <slave-hostname>
```

7. Clone the OpenShift Ansible project-repo.

```
(master)# git clone https://github.com/openshift/openshift-ansible
(master)# cd openshift-ansible
```

8. Populate the `ose-prerequisites` and `ose-install` files.

```
(master)# vi /root/openshift-ansible/inventory/byo/ose-prerequisites.yml
(master)# vi /root/openshift-ansible/inventory/byo/ose-install
```

For more information, see [ose-prerequisites.yml](#) and [ose-install](#).

9. Run the ansible-playbook to install the prerequisites on the nodes and install the atomic-openshift packages that you downloaded.



```
(master)# ansible-playbook -i inventory/byo/ose-install
inventory/byo/ose-prerequisites.yml
```

10. Run the ansible-playbook to install RedHat OpenShift.

```
(master)# ansible-playbook -i inventory/byo/ose-install
playbooks/byo/openshift_facts.yml
(master)# ansible-playbook -i inventory/byo/ose-install playbooks/byo/config.yml
```

## Installing Contrail Networking for Red Hat OpenShift

Use this procedure to install Contrail Networking for RedHat OpenShift.

1. Clone the contrail-ansible project repo.

```
(master)# mkdir contrail-ansible && cd contrail-ansible
(master)# tar -xvzf contrail-ansible*.tar.gz
(master)# mkdir playbooks/container_images && cd playbooks/container_images
```

2. Untar the contrail-kubernetes-docker-images package.

```
(master)# tar -xvzf contrail-kubernetes-docker-images_4.0.0.0-20.tgz
```

3. Populate the **inventory/hosts** file.

```
(master)# vi /root/contrail-ansible/playbooks/inventory/my-inventory/hosts
```

For more information, see [hosts](#).

4. Populate the YAML file with configuration parameters specific to your system.

```
(master)# vi
/root/contrail-ansible/playbooks/inventory/my-inventory/group_vars/all.yml
```

For more information, see [all.yml](#).

5. Run the Ansible playbook to install Contrail Networking.

```
(master)# ansible-playbook -i inventory/my-inventory site.yml
```

## Initializing Red Hat OpenShift Container and Contrail Networking

Use this procedure to initialize Red Hat OpenShift Container and Contrail Networking.

1. Create a new project and move into the project context.

```
(master)# oc login -u system:admin
(master)# oc new-project juniper
(master)# oc project juniper
```

2. Create a service account to access the APIs.

```
(master)# oc create serviceaccount useroot
```

3. Bind the service account to the role.

```
(master)# oadm policy add-cluster-role-to-user cluster-reader \
system:serviceaccount:juniper:userroot
```

4. Add the user to a privileged security context constraint.

```
(master)# oadm policy add-scc-to-user privileged
system:serviceaccount:juniper:userroot
```

5. Assign the cluster-admin role to the admin user.

```
(master)# oadm policy add-cluster-role-to-user cluster-admin admin
```

6. Assign a token to a service account.

```
(master)# oc serviceaccounts get-token userroot
```

7. Copy the service account token and log in to the contrail-kube-manager container.

```
(master)# docker ps (master)# docker exec -it contrail-kube-manager bash
```

8. Add the token to the **contrail-kubernetes.conf** file.

```
(contrail-kube-manager)# vi /etc/contrail/contrail-kubernetes.conf
[VNC]
...
token = serviceaccount-token
```

For more information, see [contrail-kubernetes.conf](#).

9. Ensure that the **cluster\_project** dict is empty.

```
(contrail-kube-manager)# vi /etc/contrail/contrail-kubernetes.conf
cluster_project = {}
```

10. Restart the contrail-kube-manager service.

```
(contrail-kube-manager)# supervisorctl -s
unix:///var/run/supervisord_kubernetes.sock
supervisor> restart all
supervisor> status
supervisor> exit
```

11. Log in to the slave node and move the OpenShift Container Network Interface (CNI) to a different location.

```
(slave)# cd /etc/cni/net.d
(slave)# mv 80-openshift-sdn.conf /etc/
```

12. Log in to the respective Web-UI dashboards.

- a. Create a password for the admin user to log in to the UI and assign permissions.

```
(master)# htpasswd /etc/origin/master/htpasswd admin
(master)# oc login -u admin
```

- b. Log in to Contrail Web-UI.

```
OpenShift Web-UI: https://<master-node-IP>:8443
Contrail Web-UI: https://<master-node-IP>:8143
```

- c. Enable Images to Run with USER in the Docker file and edit the restricted SCC.

```
(master)# oc edit scc restricted
```

- d. Change the `runAsUser.Type` strategy to `RunAsAny`.

```
runAsUser:
type: RunAsAny
```

- e. Set up BGP peering with the gateway router.

**Configure > Infrastructure > BGP Routers**

- f. Set up a network IPAM under the “default” project.

**Configure > Networking > IP Address Management > default-domain > default**

- g. Create a public virtual network.

**Configure > Networking > Networks > default-domain > default**

## Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active J-Care or Partner Support Service support contract, or are covered under warranty, and need post-sales technical support, you can access our tools and resources online or open a case with JTAC.

- JTAC policies—For a complete understanding of our JTAC procedures and policies, review the *JTAC User Guide* located at <http://www.juniper.net/us/en/local/pdf/resource-guides/7100059-en.pdf>.
- Product warranties—For product warranty information, visit <http://www.juniper.net/support/warranty/>.
- JTAC hours of operation—The JTAC centers have resources available 24 hours a day, 7 days a week, 365 days a year.

## Self-Help Online Tools and Resources

For quick and easy problem resolution, Juniper Networks has designed an online self-service portal called the Customer Support Center (CSC) that provides you with the following features:

- Find CSC offerings: <http://www.juniper.net/customers/support/>
- Search for known bugs: <https://prsearch.juniper.net/>
- Find product documentation: <http://www.juniper.net/documentation/>
- Find solutions and answer questions using our Knowledge Base: <http://kb.juniper.net/>
- Download the latest versions of software and review release notes: <http://www.juniper.net/customers/csc/software/>
- Search technical bulletins for relevant hardware and software notifications: <http://kb.juniper.net/InfoCenter/>
- Join and participate in the Juniper Networks Community Forum: <http://www.juniper.net/company/communities/>
- Open a case online in the CSC Case Management tool: <http://www.juniper.net/cm/>

To verify service entitlement by product serial number, use our Serial Number Entitlement (SNE) Tool: <https://entitlementsearch.juniper.net/entitlementsearch/>

## Opening a Case with JTAC

You can open a case with JTAC on the Web or by telephone.

- Use the Case Management tool in the CSC at <http://www.juniper.net/cm/>.
- Call 1-888-314-JTAC (1-888-314-5822 toll-free in the USA, Canada, and Mexico).

For international or direct-dial options in countries without toll-free numbers, see <http://www.juniper.net/support/requesting-support.html>.

## Revision History

---

September 2017—Revision 3, Contrail 4.0.1

June 2017—Revision 2, Contrail 4.0

June 2017—Revision 1, Contrail 4.0

Copyright © 2017 Juniper Networks, Inc. All rights reserved.

Juniper Networks, the Juniper Networks logo, Juniper, and Junos are registered trademarks of Juniper Networks, Inc. and/or its affiliates in the United States and other countries. All other trademarks may be property of their respective owners.

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.