



MIRANTIS

Mirantis OpenStack Release Notes

version 9.2

Contents

Preface	1
Intended Audience	1
Documentation History	1
Mirantis OpenStack 9.2 Release Notes	1
What's new	1
Mirantis OpenStack update procedure	1
Service decomposition	2
Network Functions Virtualization (NFV) features	2
Major components versions	3
Resolved issues	4
Fuel	4
Nova	4
Cinder	5
Neutron	5
Known issues	5
Fuel	5
Nova	6
Neutron	6
Keystone	7
Glance	7
Plugins	7
Update to minor releases (9.x)	8
Limitations	9
Prerequisites	9
Update the Fuel Master node	10
Update an existing Mirantis OpenStack environment	12
Apply customizations to a new node in Mirantis OpenStack 9.2	17
Upgrade from major releases	18
Upgrade of StackLight plugins from 0.10.x to 1.0.0	18
Upgrade of the StackLight Fuel plugins	18
Manual upgrade of InfluxDB	23

Preface

This documentation provides information on how to use Fuel to deploy OpenStack environments. The information is for reference purposes and is subject to change.

Intended Audience

This documentation is intended for OpenStack administrators and developers; it assumes that you have experience with network and cloud concepts.

Documentation History

The following table lists the released revisions of this documentation:

Revision Date	Description
February 6, 2017	9.2 GA

Mirantis OpenStack 9.2 Release Notes

Mirantis, Inc. is releasing Mirantis OpenStack version 9.2.

Mirantis OpenStack 9.2 is based on the OpenStack Mitaka release and includes enhancements, bug fixes, and known issues for the world's leading OpenStack infrastructure deployment and management technologies.

Mirantis OpenStack 9.2 is distributed as an update repository. Therefore, you can update your Mirantis OpenStack 9.0 or 9.1 to the latest version using the [Update to minor releases \(9.x\)](#) procedure.

For evaluation purposes, install Mirantis OpenStack using the Mirantis VirtualBox scripts. See [QuickStart Guide](#) for details.

What's new

Mirantis OpenStack 9.2 introduces a set of new features and enhancements.

Mirantis OpenStack update procedure

To ensure that the process of applying updates for the Mirantis OpenStack 9.x release series is reliable and scalable, Mirantis OpenStack 9.2 introduces the following improvements.

Ansible mos-updates update mechanism

Implemented the mos-updates mechanism that allows for a reliable update of an existing Mirantis OpenStack 9.0 or 9.1 environment with customizations to the version 9.2. The update procedure includes the upgrade of Linux kernel. The update of Ceph from version 0.94.6 to 0.94.9 is also available within the scope of this procedure. See [documentation](#).

Handling of an environment customizations

Added the capability to safely update a Mirantis OpenStack 9.1 environment with customizations that are not included in the Mirantis OpenStack 9.2 update.

The new update mechanism detects the customizations of installed packages and OpenStack configuration files. Apply them back after the update (except the binary files that will be lost if a new package version is available and applied). The detection of customizations in plugins is not supported.

If your customizations are included in the Mirantis OpenStack 9.2 update, they are skipped during the detection process and automatically applied to your updated Mirantis OpenStack 9.2 environment.

The update procedure ensures the integrity of customizations made before the update to Mirantis OpenStack 9.2. See [documentation](#).

Ceph update mechanism

Implemented an `update_ceph.yml` playbook mechanism that allows for a safe, automatic, zero-downtime update of Ceph installed in your Mirantis OpenStack environment 9.x. Ceph will be updated from version 0.94.6 to 0.94.9.

Service decomposition

Implemented the capability for the infrastructure operator to manage the controller node services as separate custom roles and assign these services to other physical nodes.

While planning highly loaded OpenStack environments, you can separate the controller node services, such as Neutron control plane, Keystone, MySQL, RabbitMQ, and deploy them on dedicated nodes. See [specification](#).

Network Functions Virtualization (NFV) features

Mirantis OpenStack 9.2 includes a number of features related to NFV.

NFV multiqueue support

Improved NFV workload performance with multiqueue support.

Multiqueue support enables packet sending and receiving to scale with the number of available vCPUs of the guest operating system.

Multiqueue support provides the greatest network performance benefit in the following cases:

- Traffic packets are relatively large.
- The guest operating system is active on many connections at the same time, with traffic running between guests, guest to host, or guest to an external system.
- The number of queues is equal to the number of vCPUs. Multiqueue support optimizes the RX interrupt affinity and the TX queue selection to make a specific queue private to a specific vCPU.

See [specification](#).

VXLAN support for Open vSwitch with DPDK

Added the capability to use Open vSwitch with DPDK based on VXLAN network segmentation. This enables high performance and scalable tenant networking. See [documentation](#), [specification](#).

IPv6 overlay support for Open vSwitch with DPDK

Added IPv6 overlay support for Open vSwitch with enabled DPDK.

You can use IPv6 in tenant networks when Open vSwitch with DPDK is enabled on hosts managed by Mirantis OpenStack. Consequently, you can assign an IPv6 address to a VNF, which is running on top of Open vSwitch with DPDK, in addition to an IPv4 one. See [documentation](#).

Open vSwitch Firewall Driver support

Added support for the Open vSwitch Firewall Driver.

The Open vSwitch firewall natively implements security groups as flows in Open vSwitch rather than adding an additional Linux bridge device with iptables rules per VM as in the iptables_hybrid security groups implementation. This approach increases scalability and performance.

Minimum system requirements:

- Open vSwitch 2.6 for Mirantis OpenStack with DPDK
- Open vSwitch 2.5 and Linux kernel 4.3 or newer for Mirantis OpenStack without DPDK

The Open vSwitch firewall is not enabled by default. The defaults remain the same: iptables_hybrid for Mirantis OpenStack without DPDK and no firewall for Mirantis OpenStack with DPDK. See [documentation](#), [specification](#).

Major components versions

OpenStack core projects in the Mirantis OpenStack 9.2 hardened packages support the [OpenStack Mitaka](#) release.

The following table describes the versions of software that Mirantis OpenStack 9.2 installs.

	Software	Version
Operating systems	CentOS (for the Fuel Master node only)	7.2
	Ubuntu (for the OpenStack nodes only)	14.04
Hypervisors	libvirt ^{updated}	1.3.1
	QEMU ^{updated}	2.5
Networking back end	Open vSwitch ^{updated}	2.6.1

Services	Ceph ^{updated}	0.94.9 Hammer
	Cobbler	2.6.9
	Corosync	2.3.4
	Galera	25.3.10
	HA Proxy	1.6.3
	MCollective	2.3.3
	MongoDB	2.6.10
	MySQL ^{updated}	5.6.33
	Pacemaker	1.1.14
	Puppet	3.8.5
	RabbitMQ ^{updated}	3.6.6

Resolved issues

This section lists a number of Mirantis OpenStack resolved issues sorted by OpenStack components. For a complete list, refer to [Mirantis OpenStack](#) and [Fuel for OpenStack](#) Launchpad projects.

Fuel

- Fixed the issue with the warning about a plugin being not hot pluggable that displayed in the Fuel web UI when adding a hot pluggable Fuel plugin to an existing Mirantis OpenStack environment. [LP1616825](#)
- Fixed the issue with Murano OSTF tests failing for deployments with the TLS feature enabled. [LP1590633](#)
- Fixed the issue that caused 400 Client Error when adding numbers and symbols to a new role name. [LP1625293](#)
- Fixed the issue that affected the Fuel Master node upgraded from version 8.0 to 9.1 and caused nodes to appear unavailable through MCollective. [LP1561092](#)
- Fixed the issue with File Download Error that occurred when downloading the YAML workflow for plugins using the Fuel web UI. [LP1619341](#)
- Added the capability to define arbitrary roles for the NTP server. [LP1563465](#)
- Fixed the issue that occurred after upgrading the Fuel Master node from 8.0 to 9.1. The Missing a required parameter uids error does not occur anymore when deleting a compute node from an existing Mirantis OpenStack 8.0 environment running on the Fuel Master node 9.2. [LP1628500](#)

Nova

- Fixed the issue with instances evacuation failing with the following error message: Virtual Interface creation failed. [LP1590490](#)

Cinder

- Fixed the issue with deleting a large number of volumes simultaneously. [LP1550192](#)

Neutron

- Fixed the issue with DPDK failing to initialize port and leading to the environment deployment failure if both DPDK and SR-IOV are configured on ports of the same Intel XL710 NIC. [LP1583077](#)

Known issues

This section lists the Mirantis OpenStack known issues sorted by OpenStack components.

Fuel

- While updating a Mirantis OpenStack environment to 9.2, a Deploy Changes button appears in the Fuel web UI after executing the Ubuntu kernel update playbook. Clicking this button does not perform a complete deployment. Therefore, do not use Deploy Changes until you finish the upgrade procedure through CLI. [LP1661237](#)
- During provisioning, the same IP address is provided to a new node if the old node goes offline and the DHCP lease for the IP address of this node expires, or if the DHCPREQUEST and DHCPACK packets are lost during the IP lease renewal. As a result, an environment provisioning fails. [LP1630299](#)

As a workaround, before you power on a new node, increase the DHCP lease time to the average time between the bootstrap of the first node and the end of provisioning process.

To increase the DHCP lease time:

1. Log in to the Fuel Master node.
2. Change the directory to `/etc/puppet/mitaka-9.0/modules/fuel/manifests/dnsmasq/`.
3. Open the `dhcp_range.pp` file for editing.
4. In the `define fuel::dnsmasq::dhcp_range` section, increase the value for the `lease_time` parameter. The default lease time is 120m.
5. Run the Puppet `cobbler.pp` file to apply the update:

```
puppet apply /etc/puppet/modules/fuel/examples/cobbler.pp
```

6. If the new node(s) was previously discovered, reboot the node before provisioning.
- Fuel fails to change the DataStore RegEx value on the VMware tab in the Fuel web UI. [LP1616457](#)
 - In rare cases, after a warm reboot of the primary controller node, Horizon may become unavailable through the Fuel web UI due to an incorrect routing in the HAProxy network namespace that contains the `b_public` virtual IP. [LP1657532](#)

The workaround is to restart HAProxy:

```
crm resource restart clone_p_haproxy
```

- The Fuel web UI does not detect the changes made in the VMware tab. The workaround is to make and deploy the changes using the Fuel CLI. [LP1593277](#)
- OSTF tests may fail with the following error message: vCenter: Create volume and attach it to instance because after scaling the environment, the cinder-volume service cannot establish connection to the AMQP server. The workaround is to restart the cinder-volume service. [LP1628940](#)
- In some cases, for example, after redeploying your Mirantis OpenStack environment, a Service neutron-ovs-metadata not found error may occur. As a workaround, apply the [patch](#) for manifests to your environment to /etc/puppet/modules on the Fuel Master node. [LP1658952](#)

Nova

- Affects the users who perform an upgrade from prior major releases and have instances without a configuration drive in their environment. After changing the force_config_drive parameter to True in nova.conf, you cannot restart the instance. The workaround is to cold-migrate the instance after this change. This will rebuild the configuration drive. [LP1626198](#)
- When VMware vSphere 6 is used as a hypervisor, you cannot take snapshots for instances with disk space larger than 2 GB. [LP1595206](#)

The workaround is to increase the HTTP read timeout in VMware vCenter:

1. Make a back-up copy of the the vpxd.cfg file.
2. Open the vpxd.cfg file for editing.
3. Change the readTimeoutMs setting to 600000:

```
<config>
  <vmacore>
    <http>
      <readTimeoutMs>600000</readTimeoutMs>
    </http>
  </vmacore>
  ..
</config>
```

4. Stop and restart VMware vCenter.
- In rare cases, if a VM is migrated several times in a row, SR-IOV migration may lead to the ERROR status of the instance. [LP1592409](#)

Neutron

- Affects the environments with the Open vSwitch firewall enabled and remote_group_id defined in security group rules of VMs. Booting the VMs may result in performance

degradation due to a solid number of Open vSwitch flows being generated. As a workaround, do not set the `remote_group_id` in security group rules. [LP1658711](#)

- An environment deployment with the i40e driver for the Intel® XL710 NIC fails with the following error message:

```
Parameter mtu failed on L23_stored_config[ens11f1]: Validate method
failed for class mtu: undefined method `to_i' for
{"value"=>{"value"=>"1504"}}:Hash at
/etc/puppet/modules/l23network/manifests/l2/port.pp:119
```

LP1587310

Keystone

- Keystone reads users from LDAP too slowly. As a workaround, use filters for users in `keystone.conf`. The fix that implements limiting the number of fetched users will be delivered in one of the future OpenStack releases. [LP1496840](#)

Glance

- Adding a custom field to a Glance image through Horizon results in the `nova image-show` command failing with the 500 BadGateway error. [LP1635241](#)

Plugins

- The DPDK vRouter of Contrail version 3.1 or 3.2 does not work on Linux kernel 4.4. Only the environments with the Fuel Contrail plugin as a networking driver and with DPDK enabled are affected. The symptom is that the DPDK-vRouter process does not start. To verify, run `contrail-status` on the compute node. The issue leads to the compute node not functioning. The VMs that were already allocated have non-functioning networking, new VMs fail to start. The compute nodes without DPDK operate correctly. [LP1649889](#)

The workaround is to downgrade kernel on the affected compute nodes to 3.13.0-85-generic:

```
apt-get install linux-image-3.13.0-85-generic
edit /etc/default/grub
update-grub
```

To update an environment with the Contrail plugin and without DPDK to 9.2:

1. Update the Contrail plugin to 5.0-5.0.1:

```
fuel plugins --update /tmp/contrail-5.0-5.0.1-1.noarch.rpm
```

2. Prepare the Contrail plugin local repository:

1. Copy the [Juniper Contrail installation package](#) to the Fuel Master node.

2. Run the installation script to unpack the vendor package and populate the plugin repository:

```
scp contrail-install-packages_3.0.2.1-4~liberty_all.deb \  
<Fuel Master node ip>:/var/www/nailgun/plugins/contrail-5.0/  
ssh <Fuel Master node ip> /var/www/nailgun/plugins/contrail-5.0/install.sh
```

3. Update your environment as described in [Update an existing Mirantis OpenStack environment](#).

Note

After updating the Contrail plugin to 5.0-5.0.1, it is not possible to redeploy an environment until you finish updating of your Mirantis OpenStack to 9.2.

It is not possible to update a 9.1 environment with DPDK to 9.2. However, with the Fuel Contrail plugin version 5.0-5.0.1 and kernel 3.13, you can create a new 9.2 environment both with and without DPDK enabled.

- The Contrail plugin for Fuel is not compatible with the Murano plugin for Fuel. [LP1645795](#)
- Due to an issue in the StackLight InfluxDB-Grafana plugin for Fuel, the plugin has to be updated to version 0.10.3 prior to updating your Mirantis OpenStack to 9.2. [LP1652640](#)

Update to minor releases (9.x)

Mirantis OpenStack 9.2 is distributed as an update repository. With the help of Mirantis Support, you can safely update your Mirantis OpenStack 9.0 or 9.1 to 9.2 using the procedure described in this section.

During the Mirantis OpenStack update, the OpenStack services are updated as well as RabbitMQ and MySQL. An important part of the update procedure is the Linux kernel upgrade to version 4.4. This procedure also contains the update of Ceph from version 0.94.6 to 0.94.9. The update of other environment components does not occur within the scope of Mirantis OpenStack update.

Important

Updating of a Mirantis OpenStack deployment results in a downtime of the entire environment. Therefore, before applying the updates to production, you must plan a maintenance window and back up your deployment as well as test the updates on your staging environment.

- We strongly recommend contacting Mirantis Support if you plan to update your Mirantis OpenStack environment.
- The update procedure must be executed as a single flow with no delays between steps.

This section includes the following topics:

Limitations

The Mirantis OpenStack 9.0 or 9.1 to 9.2 update procedure has the following limitations:

- Only Ubuntu-based Mirantis OpenStack update is supported.
- The files of patches generated by the `mos-mu` playbook have absolute target paths. Therefore, if you want to apply your own patch during the update, this patch should also have an absolute path.
- The detection of customizations in plugins is not supported.
- The update procedure was only tested with the following Fuel plugins enabled: StackLight, Contrail, Murano, and LDAP.
- If your environment has the StackLight InfluxDB-Grafana plugin for Fuel, update the plugin to version 0.10.3 before updating your Mirantis OpenStack to 9.2.
- If your environment has the Juniper Contrail Networking plugin for Fuel, update the plugin to version 5.0.1 before updating your Mirantis OpenStack to 9.2. See [Plugins release notes](#) for details.

Prerequisites

Caution!

Carefully read the whole [Update to minor releases \(9.x\)](#) section to understand the update scenario.

- We strongly recommend contacting Mirantis Support if you plan to update your Mirantis OpenStack environment.
- The update procedure must be executed as a single flow with no delays between steps.

Before you update Mirantis OpenStack from version 9.0 or 9.1 to 9.2, verify that you have completed the following tasks:

1. Read the [Limitations](#) section.
2. Test the update instructions in a lab environment before applying the updates to production.
3. Plan a maintenance window before applying the updates to production, as some update steps result in a downtime for existing workloads.
4. Back up the Fuel Master node as described in the [Fuel User Guide](#).
5. Verify that you have Internet access on the Fuel Master node to download the updated repository.
6. Verify that you have about 2.5 GB of free space in the `/var/www/nailgun` folder if you want to store the updates repository locally and obtain it with the `fuel-mirror` tool.
7. If the Fuel default admin password was changed, update it in the `FUEL_ACCESS:password` section of the `/etc/fuel/astute.yaml` file.
8. Log in to the Fuel Master node CLI as root.
9. Add the `mos92-updates` repository:

```
yum install -y http://mirror.fuel-infra.org/mos-repos/centos/mos9.0-centos7/9.2-updates/x86_64/Packag
```

- 10 Clean the YUM cache:

```
yum clean all
```

- 11 Install the `mos-updates` package:

```
yum install -y mos-updates
```

After completing these tasks, proceed to [Update the Fuel Master node](#).

Update the Fuel Master node

Caution!

Carefully read the whole [Update to minor releases \(9.x\)](#) section to understand the update scenario.

- We strongly recommend contacting Mirantis Support if you plan to update your Mirantis OpenStack environment.

- The update procedure must be executed as a single flow with no delays between steps.

Before creating a new environment, update the Fuel Master node from version 9.0 or 9.1 to 9.2 using the procedure below. You can also use this procedure to apply the latest maintenance updates to the Fuel Master node of your existing Mirantis OpenStack 9.2.

If you already have a Mirantis OpenStack 9.0 or 9.1 environment, before updating the Fuel Master node, take a note that all the environment customizations will be lost during the update to version 9.2. Therefore, use the customizations detection reports created before an environment update to make a decision on whether it is worth proceeding with the update. For details, see [Update an existing Mirantis OpenStack environment](#).

To update the Fuel Master node:

1. Verify that you have completed the tasks described in [Prerequisites](#).
2. Log in to the Fuel Master node CLI as root.
3. Change the directory to `mos_playbooks/mos_mu/`.
4. Perform a preparation playbook for the Fuel Master node. The playbook installs and prepares necessary tools for the update. Also, it restarts the `astute` and `nailgun` services.

```
ansible-playbook playbooks/mos9_prepare_fuel.yml
```

5. Update the Fuel Master node packages, services, and configuration:

```
ansible-playbook playbooks/update_fuel.yml -e '{"rebuild_bootstrap":false}'
```

Warning

During the update procedure, the Fuel Master node services will be restarted automatically.

6. Upgrade the Ubuntu kernel to version 4.4 for the Fuel bootstrap.

Note

Skip this step if you already have the latest Mirantis OpenStack version (9.2) and only apply the maintenance update.

```
ansible-playbook playbooks/mos9_fuel_upgrade_kernel_4.4.yml
```

7. Verify that the Fuel Master node is successfully updated to version 9.2:

- In the Fuel web UI, verify the version number in the bottom left corner of the page.
- In the Fuel CLI, run `fuel2 fuel-version`. The output of the command should be as follows:

```
# fuel2 fuel-version
openstack_version: mitaka-9.0
release: '9.2'
```

After completing these steps, proceed to [Update an existing Mirantis OpenStack environment](#) if you have a Mirantis OpenStack environment 9.0 or 9.1.

Update an existing Mirantis OpenStack environment

Caution!

Carefully read the whole [Update to minor releases \(9.x\)](#) section to understand the update scenario.

- We strongly recommend contacting Mirantis Support if you plan to update your Mirantis OpenStack environment.
- The update procedure must be executed as a single flow with no delays between steps.

If you have a running Mirantis OpenStack 9.0 or 9.1 environment, you can safely update it to the 9.2 version. You can also use this procedure to apply the latest maintenance updates to your existing Mirantis OpenStack 9.2.

During the update procedure, you create and assess the reports of your environment customizations, if any. This helps you make a decision on whether it is worth proceeding with the update since some customizations can be lost during the update process. Be aware that the detection of customizations in plugins is not supported.

To update an existing Mirantis OpenStack environment to 9.2:

1. Verify that you have updated the Fuel Master node as described in [Update the Fuel Master node](#).
2. Log in to the Fuel Master node CLI as root.
3. Change the directory to `mos_playbooks/mos_mu/`.

Note

If you want to use the fuel-mirror tool to obtain local copies of Ubuntu and mos repositories, run fuel-mirror before the next step. Otherwise, the online repositories will be installed by an environment preparation playbook.

4. Perform the preparation playbook for the environment. The playbook adds the update repository to each node of the environment, configures the /etc/apt/preferences.d/ folder, updates and restarts MCollective.

```
ansible-playbook playbooks/mos9_prepare_env.yml -e '{"env_id":<ENV_ID>}'
```

5. Run the environment configuration check using Noop run to simulate the changes and verify that the update does not override the important customizations of your environment.

Note

If your environment does not contain customizations, skip to step 9.

```
fuel2 env redeploy --noop <ENV_ID>
```

It may take a while for the task to complete. When the task succeeds, its status changes from running to ready.

To verify the task status, run fuel2 task show <TASK_ID>. The task ID is specified in the output of the configuration check command.

6. Verify the summary of the configuration check task:

```
fuel2 report <TASK_ID>
```

The task ID is specified in the output of the fuel2 task list command. The name of the task is dry_run_deployment.

The detailed Noop run reports are stored on each OpenStack node in the /var/lib/puppet/reports/node-FQDN/timestamp.yaml directory.

Warning

Some configuration and architecture customizations of the environment can be lost during the update process. Therefore, use the customizations detection reports made by Noop run to make a decision on whether it is worth proceeding with the update.

7. Collect the Python OpenStack code customizations:

```
ansible-playbook playbooks/gather_customizations.yml -e '{"env_id":<ENV_ID>}'
```

Caution!

If a Python OpenStack package was customized by adding a new file, such file will not be detected.

You may use flags to manage this procedure. For example:

- "health_check":false to skip the health checks task
- "apt_update":true to repeat the generation of APT files
- "gather_customizations":true to repeat gathering of customizations
- "md5_check":true to repeat MD5 hash verification

Example:

```
ansible-playbook playbooks/gather_customizations.yml -e \
'{"env_id":<ENV_ID>,"gather_customizations":true}'
```

Sometimes, playbooks may fail, for example, when a customized package was installed not from the mos repository. Modifying specific flags may resolve the issue. But use the flags with caution.

Note

If you have some other patches that should be applied to this environment, you can manually add these customizations to the /fuel_mos_mu/env_id/patches/ folder on the Fuel Master node. Add the ID prefix to every patch name, such as 0x-patch_name, 0y-patch_name, starting from the 01- prefix.

After the update, you can use this folder to apply new customizations.

8. Verify that the customizations in the OpenStack packages are the same on all nodes. Also, verify that the customizations are applied correctly to new versions of packages. Use the following command:

```
ansible-playbook playbooks/verify_patches.yml -e '{"env_id":<ENV_ID>}'
```

The output contains:

1. The consistency verification of the OpenStack packages. The customization for same package should be the same on all nodes. For example, the python-nova package should have the same 0 patch ID on every node:

```
TASK [Show results of customizations consistency Verification] *****
ok: [node-3.test.domain.local] => {
  "msg": [
    "Legenda:",
    " '-' - no patch (customization) for the package on this node",
    " 'x' - ID of patch",
    "",
    "nodes/packages python-nova",
    "node-1      0",
    "node-2      0",
    "node-3      0"
  ]
}
```

2. The result of the customizations applied to the updated versions of the OpenStack packages:

```
TASK [Show results of Patches Verification] *****
ok: [node-1.domain.tld] => {
  "msg": [
    "",
    "----- ./00-customizations/python-neutron_customization.patch",
    "patching file usr/lib/python2.7/dist-packages/neutron/__init__.py",
    "[OK]  python-neutron is customized successfully",
    "",
    "----- ./00-customizations/python-nova_customization.patch",
    "patching file usr/lib/python2.7/dist-packages/nova/__init__.py",
    "[OK]  python-nova is customized successfully"
  ]
}
```

9. If Ceph is installed on your environment, update Ceph from version 0.94.6 to 0.94.9.

Note

Skip this step if you already have the latest Mirantis OpenStack version (9.2) and only apply the maintenance update.

```
ansible-playbook playbooks/update_ceph.yml -e '{"env_id":<env_id>,"add_ceph_repo":true}'
```

Caution!

Do not restart the nodes at this stage. All nodes will be restarted in step 13.

10 Update the environment:

```
fuel2 update --env <ENV_ID> install --repos mos9.2-updates
```

To verify the update progress, run `fuel2 task show <TASK_ID>`. The task ID is specified in the output of the `fuel2 update install` command.

11 Upgrade the Ubuntu kernel to version 4.4.

Note

Skip this step if you already have the latest Mirantis OpenStack version (9.2) and only apply the maintenance update.

```
ansible-playbook playbooks/mos9_env_upgrade_kernel_4.4.yml -e '{"env_id":<ENV_ID>}'
```

Caution!

Do not restart the nodes at this stage. All nodes will be restarted in step 13.

12 Apply the customizations (if any) accumulated in `/fuel_mos_mu/env_id/patches` to your . updated environment:

```
ansible-playbook playbooks/mos9_apply_patches.yml -e '{"env_id":<ENV_ID>,"restart":false}'
```

13 Restart all nodes of your environment to apply the Ubuntu kernel upgrade as well as . updates for non-OpenStack services (such as RabbitMQ, MySQL, Ceph). The restart order is as follows:

1. The controller nodes restart.
2. If present, Ceph monitors stop.
3. The remaining nodes restart.
4. The system is waiting until all Ceph OSDs are up, if present.
5. If present, Ceph monitors start.

Warning

This step assumes a major downtime of the entire environment that may last up to several hours depending on your environment.

Run the following command:

```
ansible-playbook playbooks/restart_env.yml -e '{"env_id":<ENV_ID>}'
```

- 14 Verify that your environment is successfully updated to the latest version of Mirantis OpenStack 9.2:

```
ansible-playbook playbooks/get_version.yml -e '{"env_id":<ENV_ID>}'
```

Example of the system response fragment:

```
TASK [Show current MU] *****
ok: [node-1.test.domain.local] => {
  "msg": [
    "9.2"
  ]
}
```

- 15 On every Mirantis OpenStack node, verify that the Ubuntu kernel is successfully upgraded to version 4.4:

```
uname -r
```

Seealso

[Apply customizations to a new node in Mirantis OpenStack 9.2](#)

Apply customizations to a new node in Mirantis OpenStack 9.2

Once you update your Mirantis OpenStack environment to version 9.2, you may need to further modify it, for example, by adding a new node. In this case, you may want to apply the customizations of your existing environment (if any) to this node. Be aware that when you add a new node to a Mirantis OpenStack 9.2 environment, it already contains the upgraded Ubuntu kernel version 4.4.

Before you proceed with applying customizations, redeploy your existing Mirantis OpenStack 9.2 environment with a newly added node.

To apply customizations to a new node:

1. Log in to the Fuel Master node CLI.
2. Change the directory to `mos_playbooks/mos_mu/`.
3. Apply the customizations of your existing environment to the new node:

```
ansible-playbook playbooks/mos9_apply_patches.yml -e '{"env_id":<ENV_ID>}' --limit <NODE_FQDN>
```

This task restarts the OpenStack services on the node to apply customizations.

See also

[Update an existing Mirantis OpenStack environment](#)

Upgrade from major releases

You can upgrade the Fuel Master node 8.0 (Liberty) to Mitaka using the backup-restore procedure that involves the reinstallation of the Fuel Master node. See [documentation](#), [spec](#).

After the upgrade, update the Fuel Master node to the latest Mitaka. See [documentation](#).

Caution!

Currently, a Mirantis OpenStack environment upgrade from 8.0 to 9.2 is not supported. But you can use the upgraded Fuel Master node 9.2 to maintain your old environments as well as create new Mirantis OpenStack 9.2 environments.

We strongly recommend contacting Mirantis Support if you plan to upgrade your Mirantis OpenStack environment or the Fuel Master node.

Upgrade of StackLight plugins from 0.10.x to 1.0.0

This section provides a step-by-step instruction to upgrade the StackLight Fuel plugins from version 0.10.x to 1.0.0 after you update your Mirantis OpenStack to version 9.2.

This section includes the following topics.

Upgrade of the StackLight Fuel plugins

This instruction assumes that you already have a running Mirantis OpenStack environment 9.2 with the following StackLight plugins deployed:

- Elasticsearch-Kibana
- InfluxDB-Grafana
- LMA Collector

- LMA Infrastructure Alerting

Caution!

During the upgrade, the following services will be unavailable:

- InfluxDB
- Grafana
- Nagios
- All metric_collector and log_collector services will be restarted

While those service interruptions are expected, you should not have data loss.

To upgrade StackLight Fuel plugins:

1. In the [Fuel plugins catalog](#), select the following items in the drop-down menus:
 - The MOS 9.2 Mirantis OpenStack version
 - The MONITORING category
2. Download the RPM files of the StackLight Fuel plugins.
3. Log in to the Fuel Master node.
4. Copy the downloaded RPM files to the Fuel Master node:

```
scp *1.0-1.0.0-1.noarch.rpm root@<FUEL_MASTER_NODE_IP>:
```

5. Install the plugins using the [Fuel plugins CLI](#):

```
fuel plugins --install lma_collector-1.0-1.0.0-1.noarch.rpm
fuel plugins --install influxdb_grafana-1.0-1.0.0-1.noarch.rpm
fuel plugins --install elasticsearch_kibana-1.0-1.0.0-1.noarch.rpm
fuel plugins --install lma_infrastructure_alerting-1.0-1.0.0-1.noarch.rpm
```

6. Verify that the plugins are successfully installed:

```
fuel plugins
```

Example of system response:

```
id | name                | version | package_version
---+-----+-----+-----
 9 | lma_collector        | 0.10.3 | 4.0.0
 8 | influxdb_grafana    | 0.10.3 | 4.0.0
 7 | elasticsearch_kibana | 0.10.2 | 4.0.0
10 | lma_infrastructure_alerting | 0.10.3 | 4.0.0
17 | lma_collector        | 1.0.0  | 4.0.0
```

```
15 | elasticsearch_kibana      | 1.0.0 | 4.0.0
16 | influxdb_grafana          | 1.0.0 | 4.0.0
18 | lma_infrastructure_alerting | 1.0.0 | 4.0.0
```

7. Log in to the Fuel web UI.
8. In the Settings tab, select the Other category.

Important

After you select new plugins versions, Fuel resets all plugins settings to the default values. It is mandatory that new plugins settings remain identical to the existing ones. You must report all existing settings to the newly installed plugins versions, especially user names and passwords.

The environment attributes should be saved using the `fuel env --env <ID> --download --attributes` command in the Fuel CLI.

9. In the Versions section, select 1.0.0 for the LMA collector plugin.

Note

If you cannot select new versions of plugins, reload the Fuel web UI.

- 10 Report the previous Environment label setting to the new version of LMA collector plugin.

Note

The Alerting by SMTP has been removed and is not longer supported. We recommend installing the LMA infrastructure plugin in-place.

- 11 In the Versions section, select 1.0.0 for the Elasticsearch-Kibana plugin.

12 Report all previous settings to the new version of the Elasticsearch-Kibana plugin:

- Retention period
- JVM heap size
- User name
- User password
- Enable TLS for Kibana
- Use LDAP for Kibana authentication

13 In the Versions section, select 1.0.0 for the InfluxDB-Grafana plugin.

14 Report all previous settings to the new version of the InfluxDB-Grafana plugin:

- Retention period
- Root password
- Database name
- User name (InfluxDB user)
- User password (InfluxDB user)
- User name (Grafana admin user)
- User password (Grafana admin user)
- MySQL Settings
 - MySQL database
 - MySQL username
 - MySQL password
- Enable TLS for Grafana
- Use LDAP for Grafana authentication

15 In the Versions section, select 1.0.0 for the LMA Infrastructure Altering plugin.

16 Report all previous settings to the new version of the LMA Infrastructure Altering plugin:

- Nagios HTTP password
- Receive CRITICAL, WARNING, UNKNOWN, RECOVER notifications by email
- The recipient email address
- The sender email address
- External SMTP server and port
- SMTP authentication settings:
 - Method
 - SMTP password
- Enable TLS for Nagios
- Use LDAP for Nagios authentication

17 Click Save settings.

18 Manually upgrade InfluxDB as described in [Manual upgrade of InfluxDB](#).

19 Manually upgrade Grafana as described in [Manual upgrade of Grafana](#).

20 If Nagios is deployed on several nodes, run the following commands on one Nagios node to stop the Pacemaker resources related to Nagios:

```
crm resource stop nagios3
crm resource stop apache2-nagios
```

This step prevents a race condition that causes a deployment failure.

21 On the Fuel Master node, identify the environment ID. For example:

```
fuel env
id | status   | name | release_id
---+-----+-----+-----
2 | operational | SL   | 2
```

22 Connect to the Nailgun database.

Caution!

By default, the StackLight Fuel plugins 1.0.0 use a new VIP located on the public network to expose dashboards (Kibana, Grafana, and Nagios). Because of the Fuel [limitation](#), the new dashboard links are not updated. To update the new dashboard links in the Fuel web UI, you must clean the Nailgun database before running the upgrade.

23 Delete the current dashboard links. For example:

```
sudo -u postgres psql nailgun
psql (9.3.10)
Type "help" for help.
nailgun=# delete from cluster_plugin_links where cluster_id = 2 and title in \
('Kibana', 'Grafana', 'Nagios', 'Kibana (Admin role)', 'Kibana (Viewer role)');\
DELETE 3
```

Note

The new dashboard links will be created after you complete this upgrade procedure.

24 In the Fuel web UI, open the Dashboard tab and click Deploy Changes.

· If the deployment fails at the very end of deployment:

1. On the node that fails, verify that no puppet apply process is still running:

```
ps aux|grep puppet|grep apply
```

2. If the process is running, stop it:

```
kill -9 <PID>
```

3. Redeploy the changes using the Fuel web UI.

25 Verify that the services are successfully installed as described in the plugins . documentation:

- [Elasticsearch-Kibana](#)
- [InfluxDB-Grafana](#)
- [LMA Collector](#)
- [LMA Infrastructure Alerting](#)

Manual upgrade of InfluxDB

The InfluxDB-Grafana plugin 1.0.0 introduces breaking changes with InfluxDB. For more details, see [release notes](#).

Caution!

Before you proceed with the instruction below, complete steps 1-17 of the [Upgrade of the StackLight Fuel plugins](#) procedure.

Important

The instruction below must be performed on all nodes that run the Influxdb service.

To upgrade InfluxDB:

1. Log in to the first node that runs the InfluxDB service.
2. Backup the /influxdb/meta directory:

```
cp -r /var/lib/influxdb/meta ~/influxdb-meta-0.11.backup
```

3. Identify the current IP on which InfluxDB is binded on:

```
netstat -pan|grep 8088|grep influx|grep LISTEN
```

Example of system response:

```
tcp 0 0 <IP>:8088 0.0.0.0:* LISTEN 16447/influxd
```

4. Back up the InfluxDB metadata:

```
influxd backup -host <IP>:8088 /tmp/influxdb-backup/
```

Example of system response:

```
2017/01/25 14:45:32 backing up metastore to /tmp/influxdb-backup/meta.00  
2017/01/25 14:45:32 backup complete
```

5. Stop InfluxDB:

```
/etc/init.d/influxdb stop  
influxdb process was stopped [ OK ]
```

6. Verify the InfluxDB status:

```
/etc/init.d/influxdb status
influxdb Process is not running [ FAILED ]
```

7. Back up the current `/etc/influxdb/influxdb.conf` file:

```
cp /etc/influxdb/influxdb.conf ~/influxdb.conf.0.11
```

8. Remove the obsolete influxdb startup parameters (clustering-related):

```
rm -f /etc/default/influxdb
```

9. Repeat the steps above on each of the remaining nodes that run the InfluxDB service.

- 10 On the Fuel Master node, open the `/var/www/nailgun/plugins/influxdb_grafana-1.0/repositories/ubuntu/` directory.

- 11 In this directory, find and copy the `influxdb_1.1.1_amd64.deb` package.

- 12 Install the influxdb 1.1 package on the first Influxdb node.

```
Warning
Do not restart the service at this stage.
```

```
dpkg --force-confnew -i /tmp/influxdb_1.1.1_amd64.deb
```

Example of system response:

```
(Reading database ... 92355 files and directories currently installed.)
Preparing to unpack /tmp/influxdb_1.1.1_amd64.deb ...
Unpacking influxdb (1.1.1-1) over (0.11.1-1) ...
Setting up influxdb (1.1.1-1) ...
```

```
Configuration file '/etc/influxdb/influxdb.conf'
==> Modified (by you or by a script) since installation.
==> Package distributor has shipped an updated version.
==> Using new file as you requested.
Installing new version of config file /etc/influxdb/influxdb.conf ...
System start/stop links for /etc/init.d/influxdb already exist.
```

- 13 Verify that the influxdb 1.1 package is successfully installed:

```
dpkg -l|grep influxdb
```

Example of system response:

```
ii influxdb 1.1.1-1 amd64 Distributed time-series database.
```

14 Restore the metadata with the new binary influxd:

```
influxd restore -metadir=/var/lib/influxdb/meta/ /tmp/influxdb-backup/  
Using metastore snapshot: /tmp/influxdb-backup/meta.00
```

15 Start InfluxDB:

```
/etc/init.d/influxdb start
```

Example of system response:

```
Starting influxdb...  
influxdb process was started [ OK ]
```

16 Repeat the steps 12-15 on each of the remaining nodes that run the Influxdb service.

17 Proceed to upgrading Grafana as described in [Manual upgrade of Grafana](#).

Seealso
[Official InfluxDB documentation](#)

Manual upgrade of Grafana

The InfluxDB-Grafana plugin 1.0.0 introduces support for Grafana version 4.1. Manual upgrade is required on all nodes that run the Grafana service.

Caution!

Before you proceed with the instruction below, complete steps 1-18 of the [Upgrade of the StackLight Fuel plugins](#) procedure.

Important

The instruction below must be performed on all nodes that run the Grafana server.

To upgrade Grafana:

1. Log in to the first node that runs the Grafana server.
2. Stop the Grafana server on the node:

```
/etc/init.d/grafana-server stop
* Stopping Grafana Server
...done.
```

3. On the Fuel Master node, open the `/var/www/nailgun/plugins/influxdb_grafana-1.0/repositories/ubuntu/` directory.
4. In this directory, find and copy the `grafana_4.1.1-1484211277_amd64.deb` package.
5. On the first node that runs the Grafana server, install the `grafana` package:

```
dpkg --force-confnew -i /tmp/grafana_4.1.1-1484211277_amd64.deb
```

Example of system response:

```
(Reading database ... 92094 files and directories currently installed.)
Preparing to unpack .../grafana_4.1.1-1484211277_amd64.deb ...
Unpacking grafana (4.1.1-1484211277) over (3.0.4-1464167696) ...
Setting up grafana (4.1.1-1484211277) ...

Configuration file '/etc/grafana/grafana.ini'
==> Modified (by you or by a script) since installation.
==> Package distributor has shipped an updated version.
==> Using new file as you requested.

Installing new version of config file /etc/grafana/grafana.ini ...
Installing new version of config file /etc/grafana/ldap.toml ...
Installing new version of config file /usr/lib/systemd/system/grafana-server.service ...
Processing triggers for ureadahead (0.100.0-16) ...
```

6. Verify that the Grafana server is stopped:

```
/etc/init.d/grafana-server status
* grafana is not running
```

Caution!

Do not start the Grafana server at this stage.

7. Repeat the steps above on each of the remaining nodes that run the Grafana server.

8. Proceed to step 20 of the [Upgrade of the StackLight Fuel plugins](#) instruction.