
Detach RabbitMQ plugin for Fuel Documentation

Release 1.1-1.1.1-1

Mirantis Inc.

July 08, 2016

CONTENTS

1 Document purpose	1
1.1 Detach RabbitMQ Plugin	1
1.2 Requirements	1
1.3 Limitations	1
2 Installation Guide	2
3 User Guide	3
3.1 How it works	4

DOCUMENT PURPOSE

This document provides instructions for installing, configuring and using Detach RabbitMQ plugin for Fuel.

1.1 Detach RabbitMQ Plugin

The Detach RabbitMQ plugin provides ability to install an OpenStack environment with RabbitMQ deployed on dedicated nodes. RabbitMQ consumes a lot of CPU time on big environment, and the plugin allows to separate it from other big consumers running on controller nodes like database or various OpenStack processes.

1.2 Requirements

Requirement	Version/Comment
Fuel	9.x release
OpenStack compatibility	Mitaka
Operating systems	Ubuntu 14.04 LTS

1.3 Limitations

Plugin can be enabled only in a new environment. If an environment was deployed without the plugin, it is impossible to enable and use it there later.

If the Detach RabbitMQ plugin is enabled for an environment, it is impossible to assign RabbitMQ and Controller roles to the same node.

There are non-verified plugins, which enable user to install Keystone or database on separate nodes. These plugins could be used together with Detach RabbitMQ plugin, but user should ensure that:

- either RabbitMQ and DB or Keystone roles are assigned to the same nodes
- or RabbitMQ and DB or Keystone roles are assigned to completely different nodes and no node exist, which has two of these roles assigned

For example, it is valid to assign RabbitMQ + Keystone roles to three nodes in the cluster. A bad example is environment with one RabbitMQ + Keystone node, one RabbitMQ and one Keystone node (3 nodes in total).

INSTALLATION GUIDE

1. Start with installing Fuel Master node ¹.
2. Download the plugin from Fuel Plugins Catalog ².
3. Copy the plugin on already installed Fuel Master node:

```
[user@home ~]$ scp detach-rabbitmq-1.1-1.1.1-1.noarch.rpm root@:/  
<the_Fuel_Master_node_IP>:~/
```

4. Log into the Fuel Master node. Install the plugin:

```
[root@fuel ~]# fuel plugins --install detach-rabbitmq-1.1-1.1.1-1.noarch.rpm
```

5. Verify that the plugin is installed correctly:

```
[root@fuel ~]# fuel plugins --list  
id | name | version | package_version | releases  
---+-----+-----+-----+-----  
1 | detach-rabbitmq | 1.1.1 | 3.0.0 | ubuntu (mitaka-9.0)
```

¹ <http://docs.openstack.org/developer/fuel-docs/mitaka/userdocs/fuel-install-guide.html>

² <https://www.mirantis.com/products/openstack-drivers-and-plugins/fuel-plugins/>

USER GUIDE

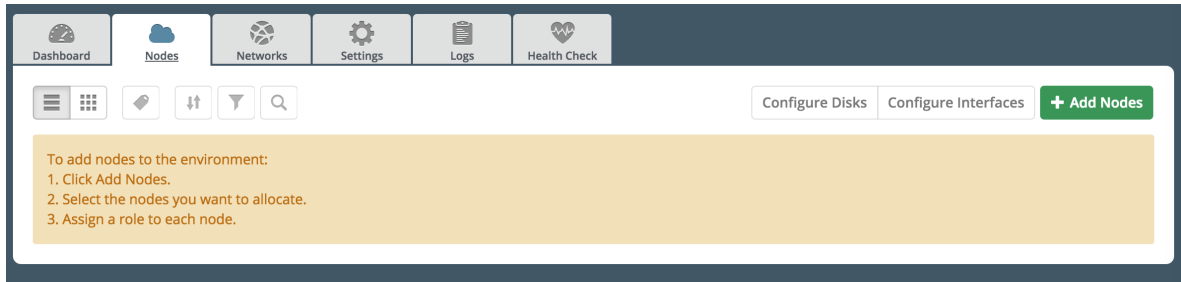
1. After the plugin is installed, [create a new OpenStack environment](#) ¹.
2. Open the Settings tab of the Fuel web UI and then select the Other menu. Select “Detach RabbitMQ Plugin” checkbox. With the radio button below select the plugin version (multiple options are available only when several versions of the plugin are installed):

OpenStack Settings

- General Detach RabbitMQ Plugin
- Security Versions 1.1.1
- Compute
- Storage
- Logging
- OpenStack Services
- Other

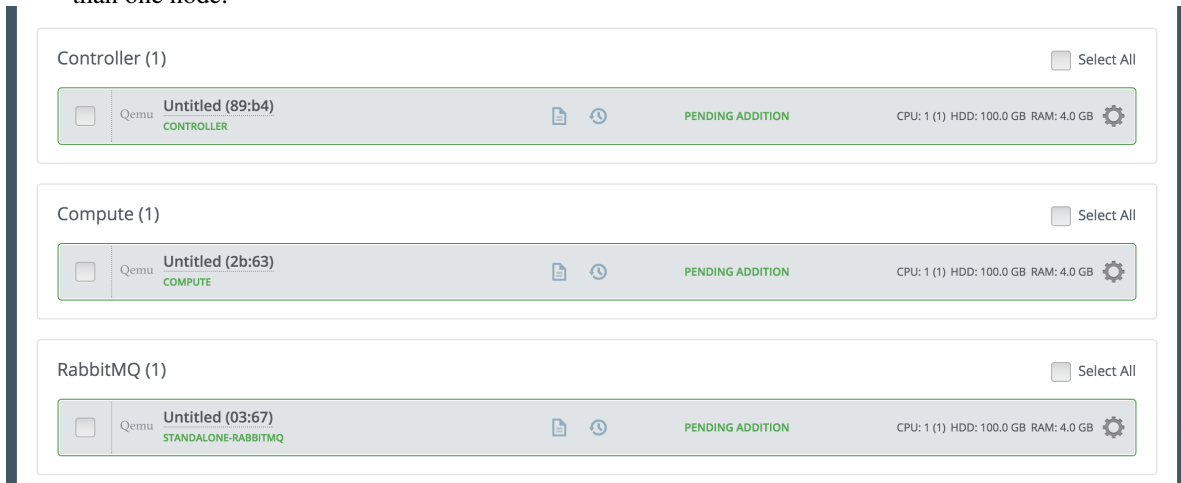
3. Go to the Nodes tab and here push Add Nodes button

¹ <http://docs.openstack.org/developer/fuel-docs/mitaka/userdocs/fuel-user-guide/create-environment.html>



Note that now RabbitMQ role is available in the roles list.

4. Add nodes to the environment with RabbitMQ role assigned to some of them. On the screenshot below you may see environment with 1 controller, 1 compute and one RabbitMQ node. You can assign RabbitMQ role to more than one node.



5. Finish configuring your environment ².
6. Deploy your environment ³.

3.1 How it works

Without the plugin, Fuel deploys RabbitMQ on Controller nodes. Here it is managed by Pacemaker along with a number of other processes.

With the plugin enabled, Fuel deploys RabbitMQ on dedicated nodes and here it is also managed by Pacemaker. But in that case RabbitMQ is not launched on the Controller nodes. Also note that two separate Pacemaker clusters are running on Controller and RabbitMQ nodes and they are not aware of each other.

When the plugin is enabled, RabbitMQ log could be found in its regular place:

- on RabbitMQ node in `/var/log/rabbitmq` directory
- on master node in `/var/log/remote/<node-name>/rabbitmq-*.log` files

The same applies to log of Pacemaker which manages RabbitMQ. Its location is:

- on RabbitMQ node `/var/log/pacemaker.log`
- on master node in the following files:

² <http://docs.openstack.org/developer/fuel-docs/mitaka/userdocs/fuel-user-guide/configure-environment.html>

³ <http://docs.openstack.org/developer/fuel-docs/mitaka/userdocs/fuel-user-guide/deploy-environment/deploy-changes.html#deploy-changes>

- /var/log/remote/<node-name>/attrd.log
- /var/log/remote/<node-name>/crmd.log
- /var/log/remote/<node-name>/cib.log
- /var/log/remote/<node-name>/lrmd.log
- /var/log/remote/<node-name>/pengine.log