
The Kafka plugin for Fuel documentation

Release 1.0.0

Mirantis Inc.

January 10, 2017

1	Overview	1
2	Installing and configuring the Kafka plugin for Fuel	3

Overview

1.1 Introduction

The Kafka Plugin installs [Apache Kafka](#) and [Apache ZooKeeper](#) in a Mirantis OpenStack environment deployed by Fuel. Apache Kafka is a publish-subscribe messaging system. It is fast, scalable, and durable.

The Kafka Plugin is created for exchanging messages between various components of StackLight and Ceilometer, but it is generic enough to accommodate other usages.

See also:

- [Kafka 0.10.0 documentation](#)

1.2 Requirements

The Kafka Plugin has the following requirements:

Requirement	Version/Comment
Mirantis OpenStack	9.x
Disk space	Provide at least 50 GB of disk space. The disk space depends on the environment size, replication settings, and data retention.
Hardware requirements	The hardware configuration (RAM, CPU, disk(s)) required by this plugin depends on the size of your cloud environment and other factors like the replication factor and retention policy for Kafka queues. The JVM Heap size is configurable through the plugin UI. Provide at least 8 GB for the JVM heap size of Kafka. For an intensive workload, you should provide 16GB of RAM for the JVM heap size of Kafka and 5 GB of RAM for the JVM heap size of Zookeeper. An average setup would require a quad-core server with at least 8 GB of RAM for Kafka and 3GB for Zookeeper and access to a 500+ IOPS disk.
OpenStack nodes	The plugin implements the <i>Kafka</i> Fuel role. The maximum number of nodes for the <i>Kafka</i> Fuel role is five. To support high-availability, the minimum number of nodes is three. You need to have an initial odd number of nodes to comply with the leader election process constraints.

1.3 Limitations

The Kafka Plugin has the following limitations:

- Kafka supports authentication, encryption, and authorization. The current version of the plugin does not support any form of security, meaning that the Kafka cluster will be “open” on the management network.

- The Kafka Plugin will not expose configuration properties for all the broker configuration parameters. That is, the Kafka broker configuration set by the plugin will not be appropriate for every usage.

1.4 References

For more information about Kafka, see:

- [Kafka plugin project at GitHub](#)
- [Official Kafka documentation](#)

Installing and configuring the Kafka plugin for Fuel

2.1 Install the plugin

Before you proceed with installing the plugin, see the *Limitations* section.

To install the Kafka Plugin:

1. Download the plugin from the [Fuel Plugins Catalog](#).
2. Copy the plugin's RPM file to the **Fuel Master node** using the secure copy **scp** command:

```
# scp fuel-plugin-ceilometer-redis/kafka-1.0-0.1.0-1.noarch.rpm /  
root@:<the_Fuel_Master_node_IP address>:/tmp
```

3. Log in to the Fuel Master node:

```
# ssh root@:<the_Fuel_Master_node_IP address>
```

4. Install the plugin:

```
[root@fuel-master ~]# cd /tmp  
[root@fuel-master ~]# fuel plugins --install kafka-1.0-0.1.0-1.noarch.rpm
```

5. Verify that the plugin is installed correctly:

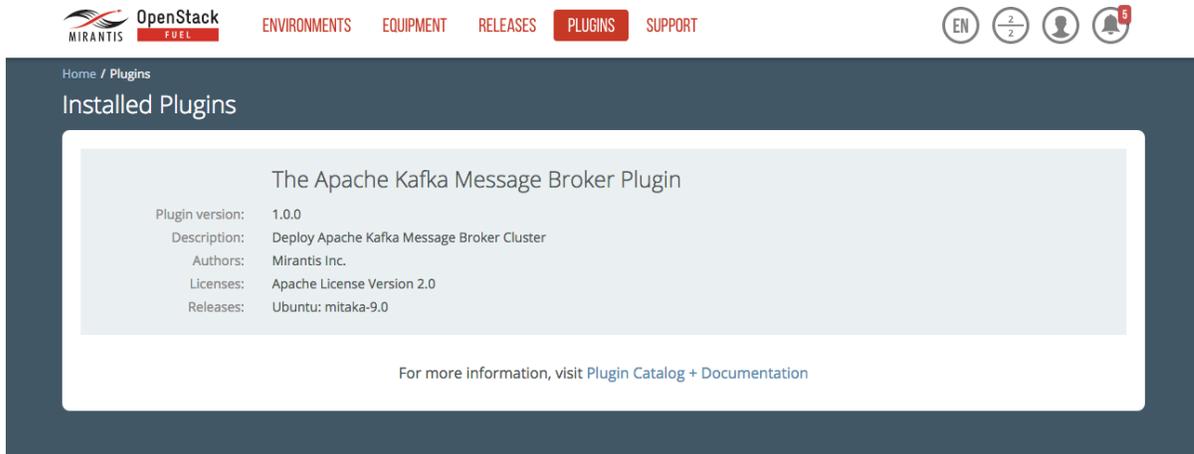
```
[root@fuel-master ~]# fuel plugins list  
id | name | version | package_version | releases  
---+-----+-----+-----+-----  
1 | kafka | 1.0.0 | 4.0.0 | ubuntu (mitaka-9.0)
```

2.2 Configure the plugin

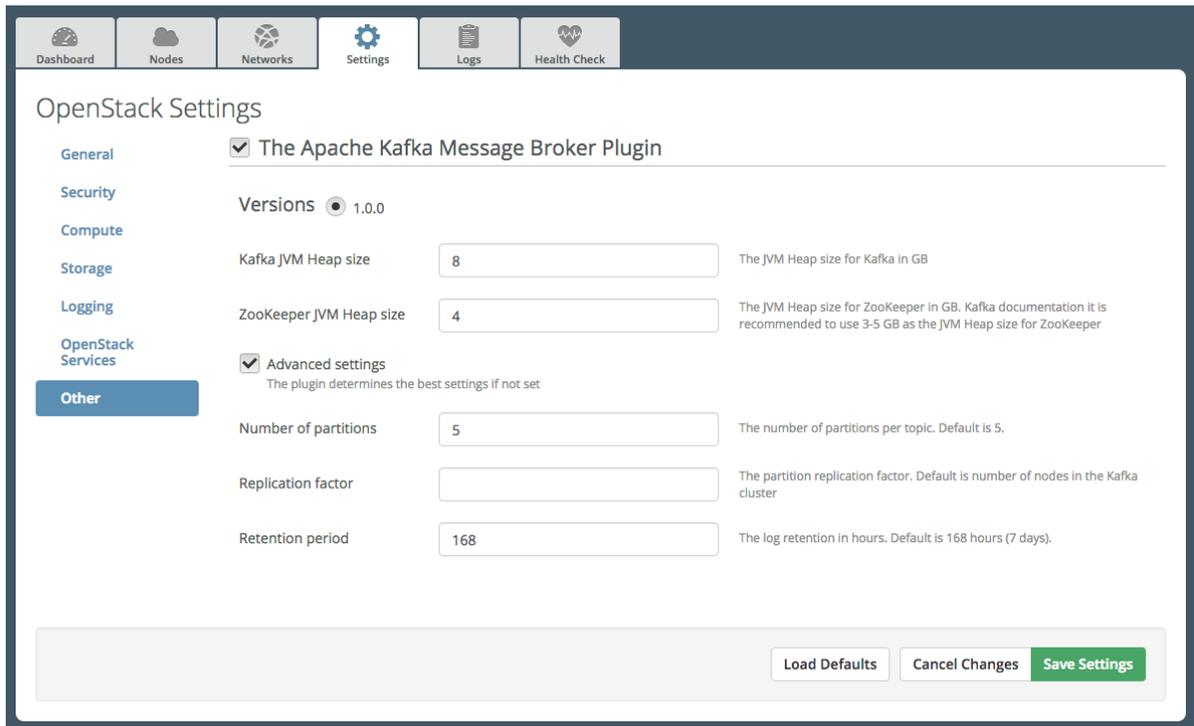
Once installed, configure the Kafka Plugin and add nodes with the **Kafka** role to a new or existing Mirantis OpenStack environment.

To configure the plugin:

1. Log in to the Fuel web UI.
2. Verify that the Kafka Plugin is listed in the *Plugins* tab:



3. Create a new environment or use an existing one.
4. To enable the plugin and configure additional settings, navigate to the *Environments* tab and select the *The Apache Kafka Message Broker Plugin* checkbox:



5. Add nodes to your environment and assign the **Kafka** role to these nodes.

Note: When adding nodes to the environment and assigning or changing a role, do not forget to use an odd number of nodes as recommended in the *Requirements* section.

Assign Roles Place the mouse cursor over a role to view description.

Base: Controller

Compute: Compute, Ironic

Storage: Cinder, Cinder Block Device, Ceph OSD

Other: Telemetry - MongoDB, Operating System, **Kafka**

Discovered (3) Select All

Qemu	Role	Status	Specs
Untitled (6d:7c)	KAFKA	DISCOVERED	CPU: 2 (2) RAM: 3.0 GB HDD: 1.0 TB
Untitled (db:bd)	KAFKA	DISCOVERED	CPU: 2 (2) RAM: 3.0 GB HDD: 1.0 TB
Untitled (6f:ab)	KAFKA	DISCOVERED	CPU: 2 (2) RAM: 3.0 GB HDD: 1.0 TB

4. Verify your network configuration.
5. Deploy your environment.

2.3 Verify the plugin

Depending on the number of nodes and deployment setup, deploying a Mirantis OpenStack environment may take 20 minutes to several hours. Once the deployment is complete, you should see a deployment success notification message.

To verify the plugin:

1. On the Fuel Master node, find the IP address of the node where Kafka is installed using the `fuel nodes` command:

```
[root@fuel ~]# fuel nodes
id|status|name                |cluster|ip   |mac   |roles
--|-----|-----|-----|----|-----|-----
1 |ready  |Untitled (fa:87) | 1      |...  |...   |kafka
2 |ready  |Untitled (12:aa) | 1      |...  |...   |kafka
3 |ready  |Untitled (4e:6e) | 1      |...  |...   |kafka
```

2. Log in to any of these nodes using SSH, for example, to node-1.
3. Run the following command:

```
root@node-1:~# netstat -ntpl | grep java
tcp6      0      0  :::9092          :::*           LISTEN    14702/java
tcp6      0      0  :::2181          :::*           LISTEN    9710/java
tcp6      0      0  :::9990          :::*           LISTEN    14702/java
```

You should see Zookeeper running and listening on port 2181 and Kafka on ports 9092 and 9990 accordingly.

4. To perform additional verification tests, for example, to try send and receive messages, see the instructions provided in the [Quick Start Guide \(steps 3-5\)](#).