Fuel StackLight Elasticsearch-Kibana Plugin Guide

Release 1.0.0

Mirantis Inc.

Jan 16, 2018

Contents

CHAPTER 1

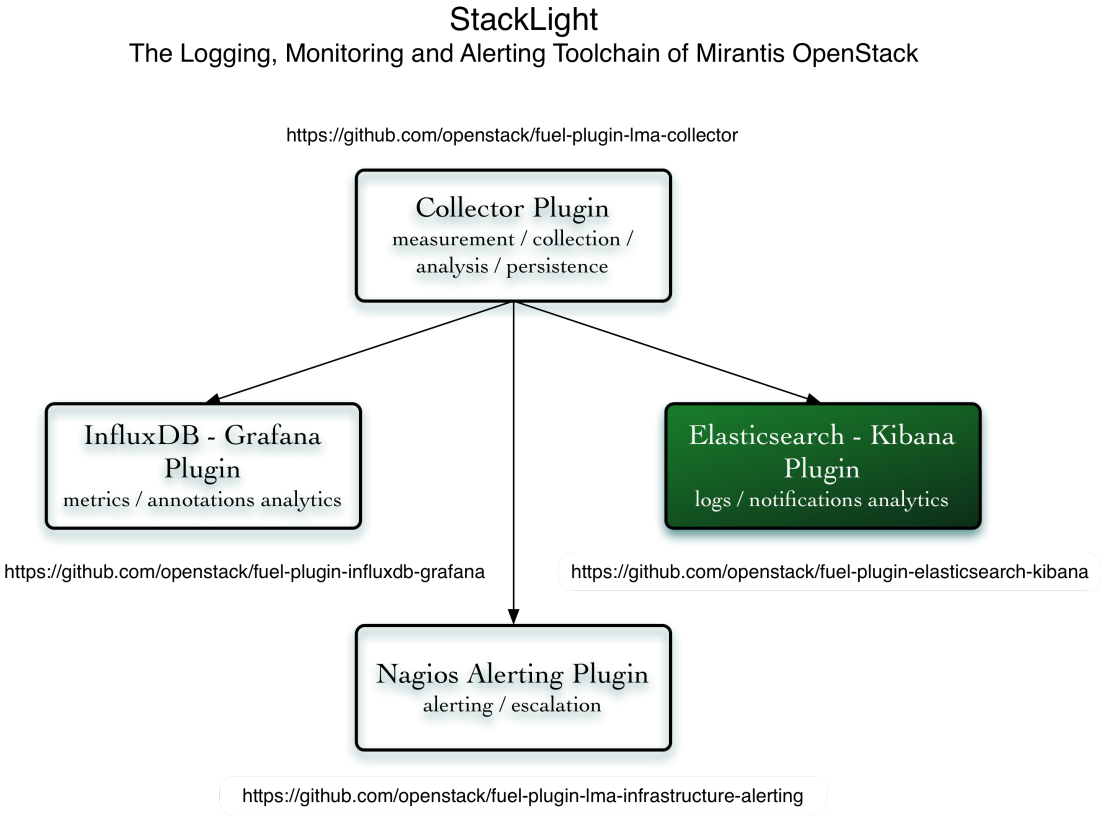
Overview

1.1 Introduction

The **StackLight Elasticsearch-Kibana plugin** is used to install and configure Elasticsearch and Kibana components that collectively provide access to the logs and notifications analytics of the so-called Logging, Monitoring, and Alerting (LMA) Toolchain of Mirantis OpenStack.

These analytics can be used to search and correlate service-affecting events which may occur on your OpenStack environment. It is an indispensable tool to troubleshoot problems.

Elasticsearch and Kibana are key components of the LMA Toolchain project, also known as StackLight:


StackLight
The Logging, Monitoring and Alerting Toolchain of Mirantis OpenStack
https://github.com/openstack/fuel-plugin-lma-collector
Collector Plugin
measurement / collection / analysis / persistence
InfluxDB - Grafana Plugin
metrics / annotations analytics
Elasticsearch - Kibana Plugin
logs / notifications analytics
https://github.com/openstack/fuel-plugin-influxdb-grafana
https://github.com/openstack/fuel-plugin-elasticsearch-kibana
Nagios Alerting Plugin
alerting / escalation
https://github.com/openstack/fuel-plugin-lma-infrastructure-alerting

1.2 Key terms

The table below lists the key terms that are used in this document.

Term	Definition
StackLight Collector	A smart monitoring agent running on every node which collects and processes the logs and the notifications of your OpenStack environment.
Elasticsearch	An open-source (Apache-licensed) application based on the Lucene search engine that makes data-like log messages easy to explore and correlate. Elasticsearch is written in Java and uses Lucene internally for all of its indexing and searching, but it aims to make full-text search easy by hiding the complexities of Lucene behind a simple, coherent RESTful API.
Kibana	An open-source (Apache-licensed) browser-based analytics and search dashboard for Elasticsearch. Kibana is easy to setup and use.

1.3 Requirements

The StackLight Elasticsearch-Kibana Plugin 1.0.0 has the following requirements:

Requirement	Version/Comment
Disk space	The plugin's specification requires: • provisioning at least 15 GB of disk space for the system • 10 GB for the logs • 30 GB for the database As a result, the installation of the plugin will fail if there is less than **55 GB** of disk space available on the node.
Mirantis OpenStack	8.0, 9.x
Hardware configuration	The hardware configuration (RAM, CPU, disk) depends on the size of your cloud environment of your cloud environment and other parameters such as the retention period and log level. A typical setup would at least requires a quad-core server with 8 GB of RAM and access to a 500-1000 IOPS disk. For sizeable production deployments it is strongly recommended to use a disk capable of 1000+ IOPS like an SSD. The actual disk space you need to run the plugin on depends on several factors including the size of your OpenStack environment, the retention period, the logging level, and workload. The more of the above, the more disk space you need to run the Elaticsearch-Kibana plugin. We also highly recommend using dedicated disk(s) for your data storage.

1.4 Limitations

The StackLight Elasticsearch-Kibana plugin 1.0.0 has the following limitations:

- Currently, the maximum size of an Elasticsearch cluster that can be installed by Fuel is limited to five nodes. But each node of an Elasticsearch cluster is configured as *master candidate* and a *storage node*. This means that each node of an Elasticsearch cluster can be selected as master, and all nodes will store data.
- The *cluster operations* may require manual operations.

1.5 Release notes

1.5.1 Version 1.0.0

The StackLight Elasticsearch-Kibana plugin 1.0.0 contains the following updates:

- Fixed an issue to allow the configuration of a list of LDAP servers. See #1624002.
- Added support to handle the scripts execution option that is required for the OpenStack Telemetry plugin.
- Fixed the curator job to work with Elasticsearch 2.x. See #1616765.
- Added support for wildcard SSL certificates. See #1608665.
- Fixed the UI issue with the LDAP protocol radio button. See #1599778.
- Fixed a race condition when scaling up the Elasticsearch cluster leading to unavailability of Kibana dashboards. #1552258.
- Prevent co-installation with the Contrail plugin. #1646550.

1.5.2 Version 0.10.0

The StackLight Elasticsearch-Kibana plugin 0.10.0 contains the following updates:

- Added support for the LDAP(S) authentication to access the Kibana UI.
- Added support for the TLS encryption to access the Kibana UI.

  To configure the TLS termination, update the plugin settings with a PEM file obtained by concatenating the SSL certificate with the private key.
- Upgraded to Elasticsearch v2.3.3.
- Upgraded to Kibana v4.5.
- Fixed the issue in logs and notifications being dropped during a long Elasticsearch outage. See #1566748.

1.5.3 Version 0.9.0

The StackLight Elasticsearch-Kibana plugin 0.9.0 contains the following updates:

- Added support for Elasticsearch and Kibana clustering for scale-out and high availability of those services.
- Upgraded to Elasticsearch 1.7.4.
- Upgraded to Kibana 3.1.3.

1.5.4 Version 0.8.0

The StackLight Elasticsearch-Kibana plugin 0.8.0 contains the following updates:

- Added support for the `elasticsearch_kibana` Fuel plugin role instead of the `base-os` role which had several limitations.
- Added support for the retention policy configuration with Elastic Curator.
- Upgraded to Elasticsearch 1.4.5.

1.5.5 Version 0.7.0

The initial release of the plugin (beta version).

1.6 Licenses

1.6.1 Third-party components

Name	Project web site	License
Elasticsearch	https://www.elastic.co/products/elasticsearch	Apache v2
Kibana	https://www.elastic.co/products/kibana	Apache v2

1.6.2 Puppet modules

Name	Project web site	License
Elasticsearch	https://forge.puppetlabs.com/elasticsearch/elasticsearch	Apache v2
Concat	https://github.com/puppetlabs/puppetlabs-concat	Apache v2
Stdlib	https://github.com/puppetlabs/puppetlabs-stdlib	Apache v2
Apache	https://github.com/puppetlabs/puppetlabs-apache	Apache v2
Firewall	https://github.com/puppetlabs/puppetlabs-firewall	Apache v2
Datacat	https://github.com/richardc/puppet-datacat	Apache v2

1.7 References

- GitHub project
- Official Kibana documentation
- Official Elasticsearch documentation

CHAPTER 2

Installing and configuring the StackLight Elasticsearch-Kibana plugin for Fuel

2.1 Install the plugin

2.1.1 Introduction

You can install the StackLight Elasticsearch-Kibana Fuel plugin using one of the following options:

- Install using the RPM file
- Install from source

The following is a list of software components installed by the StackLight Elasticsearch-Kibana Fuel plugin:

Components	Version
Elasticsearch	v2.3.3 for Ubuntu (64-bit)
Kibana	v4.5
Apache	Version coming with the Ubuntu distribution

2.1.2 Install using the RPM file

To install the StackLight Elasticsearch-Kibana Fuel plugin using the RPM file from the Fuel plugins' catalog:

1. Go to the Fuel plugins' catalog.
2. From the *Filter* drop-down menu, select the Mirantis OpenStack version you are using and the *MONITORING* category.
3. Download the RPM file.
4. Copy the RPM file to the Fuel Master node:

```
[root@home ~]# scp elasticsearch_kibana-1.0-1.0.0-0.noarch.rpm \
root@<Fuel Master node IP address>:
```

5. Install the plugin using the Fuel Plugins CLI:

```
[root@fuel ~]# fuel plugins --install elasticsearch_kibana-1.0-1.0.0-0.noarch.rpm
```

6. Verify that the plugin is installed correctly:

```
[root@fuel ~]# fuel plugins --list
id | name                   | version   | package_version
---|------------------------|-----------|----------------
1  | elasticsearch_kibana | 1.0.0    | 4.0.0
```

2.1.3 Install from source

Alternatively, you can build the RPM file of the plugin from source if, for example, you want to test the latest features of the master branch or customize the plugin.

Caution: Running a Fuel plugin that you built from source is at your own risk and is not supported.

Before you install the StackLight Elasticsearch-Kibana plugin from source, prepare an environment to build the RPM file. We recommend building the RPM file directly on the Fuel Master node not to copy that file later on.

To prepare an environment for building the plugin on the Fuel Master node:

1. Install the standard Linux development tools:

```
[root@home ~] yum install createrepo rpm rpm-build dpkg-devel
```

2. Install `pip`:

```
[root@home ~] easy_install pip
```

3. Install the Fuel Plugin Builder (the `fpb` command line) using `pip`:

```
[root@home ~] pip install fuel-plugin-builder
```

Note: You may also need to build the Fuel Plugin Builder if the package version of the plugin is higher than the package version supported by the Fuel Plugin Builder you get from `pypi`. For instructions on how to build the Fuel Plugin Builder, see the Fuel Plugin SDK Guide.

4. Clone the plugin repository:

```
[root@home ~] git clone \
  https://github.com/openstack/fuel-plugin-elasticsearch-kibana.git
```

5. Verify that the plugin is valid:

```
[root@home ~] fpb --check ./fuel-plugin-elasticsearch-kibana
```

6. Build the plugin:

```
[root@home ~] fpb --build ./fuel-plugin-elasticsearch-kibana
```

To install the plugin:

1. Once you have created the RPM file, install the plugin:

```
[root@fuel ~] fuel plugins --install \
  ./fuel-plugin-elasticsearch-kibana/*.noarch.rpm
```

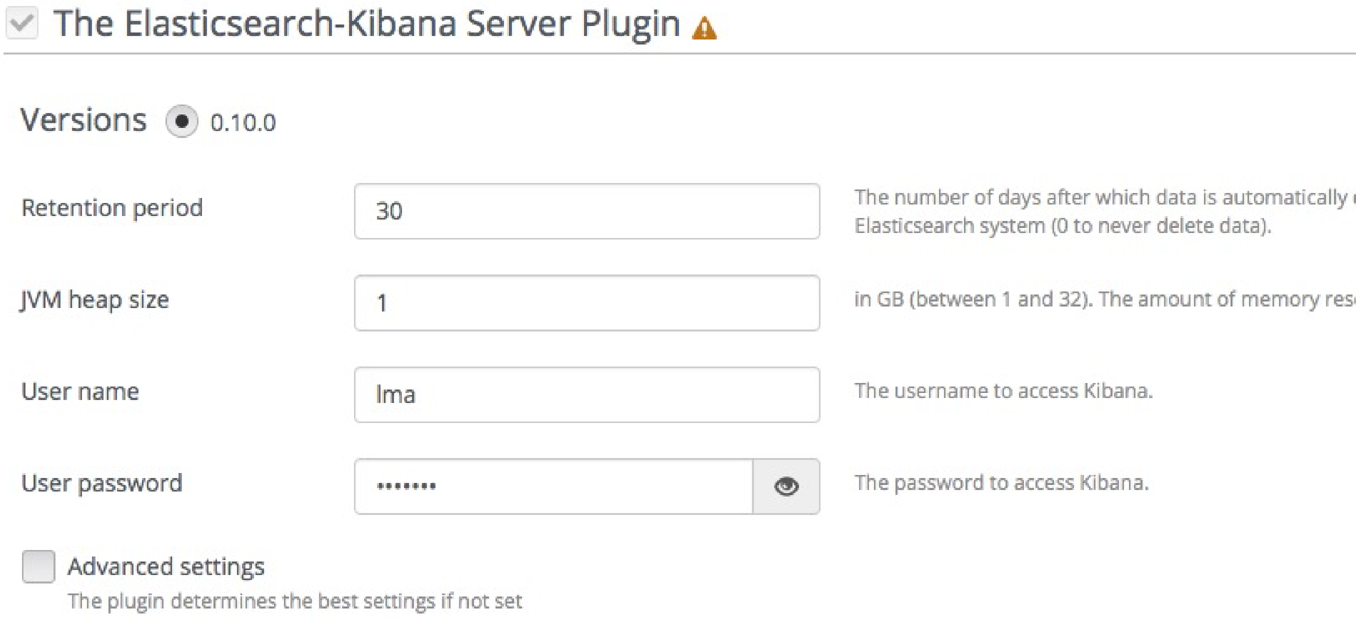
2. Verify that the plugin is installed correctly:

```
[root@fuel ~]# fuel plugins --list
id | name                  | version | package_version
---|-----------------------|---------|----------------
1  | elasticsearch_kibana | 1.0.0   | 4.0.0
```

2.2 Configure the plugin during an environment deployment

To configure the StackLight Elasticsearch-Kibana plugin during an environment deployment:

1. Using the Fuel web UI, create a new environment.
2. In the Fuel web UI, click the *Settings* tab and select the *Other* category.
3. Scroll down through the settings to find the *StackLight Elasticsearch-Kibana Plugin* section.
4. Select *StackLight Infrastructure Alerting Plugin* and fill in the required fields as follows:



☑ The Elasticsearch-Kibana Server Plugin ⚠

Versions ● 0.10.0

Retention period: 30 — The number of days after which data is automatically
Elasticsearch system (0 to never delete data).

JVM heap size: 1 — in GB (between 1 and 32). The amount of memory res

User name: lma — The username to access Kibana.

User password: ••••••• — The password to access Kibana.

□ Advanced settings
The plugin determines the best settings if not set

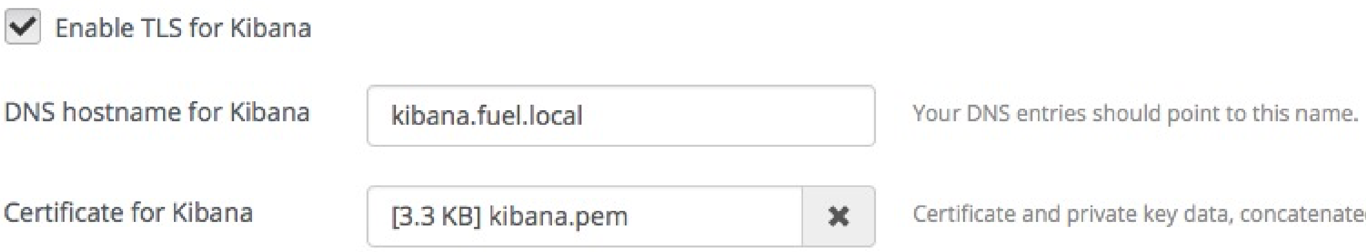
   (a) Specify the number of days to retain your data.

   (b) Specify the *JVM heap size* for Elastisearch. Use the tips below:

      - By default, 1 GB of heap memory is allocated to the Elasticsearch process. This value is enough to run Elasticsearch for local testing only.
      - To run Elasticsearch in production, allocate minimum 4 GB of memory. But we recommend allocating 50% of the available memory up to 32 GB maximum.
      - If you set a value greater than the memory size, Elasticsearch will not start.
      - Reserve enough memory for operating system and other services.

   (c) Select and edit *Advanced settings* if Elasticsearch and Kibana are installed on a cluster of nodes.

5. Select *Enable TLS for Kibana* if you want to encrypt your Kibana credentials (username, password). Then, fill in the required fields as follows:



☑ Enable TLS for Kibana

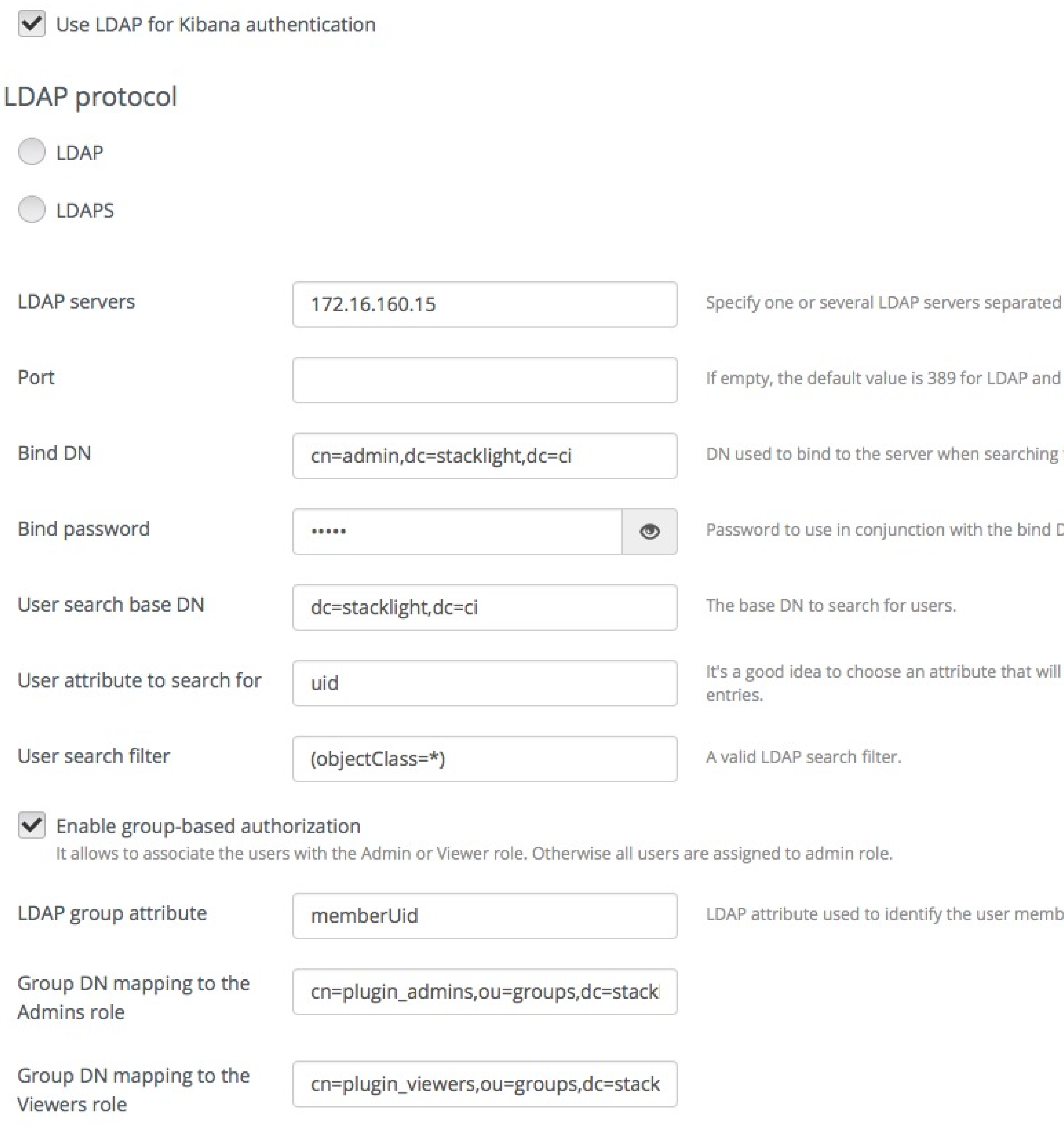
DNS hostname for Kibana: kibana.fuel.local — Your DNS entries should point to this name.

Certificate for Kibana: [3.3 KB] kibana.pem ✖ — Certificate and private key data, concatenate

(a) Specify the DNS name of the Kibana server. This parameter is used to create a link in the Fuel dashboard to the Kibana server.

(b) Specify the location of a PEM file that contains the certificate and the private key of the Kibana server that will be used in TLS handchecks with the client.

6. If you want to authenticate through LDAP to Kibana, select *Use LDAP for Kibana authentication*. Then, fill in the required fields as follows:



☑ Use LDAP for Kibana authentication

LDAP protocol

○ LDAP

○ LDAPS

LDAP servers: 172.16.160.15 — Specify one or several LDAP servers separated

Port: — If empty, the default value is 389 for LDAP and

Bind DN: cn=admin,dc=stacklight,dc=ci — DN used to bind to the server when searching

Bind password: ••••• — Password to use in conjunction with the bind D

User search base DN: dc=stacklight,dc=ci — The base DN to search for users.

User attribute to search for: uid — It's a good idea to choose an attribute that will entries.

User search filter: (objectClass=*) — A valid LDAP search filter.

☑ Enable group-based authorization

It allows to associate the users with the Admin or Viewer role. Otherwise all users are assigned to admin role.

LDAP group attribute: memberUid — LDAP attribute used to identify the user memb

Group DN mapping to the Admins role: cn=plugin_admins,ou=groups,dc=stack

Group DN mapping to the Viewers role: cn=plugin_viewers,ou=groups,dc=stack

(a) Select *LDAPS* if you want to enable LDAP authentication over SSL.

(b) Specify one or several *LDAP servers* addresses separated by space. Those addresses must be accessible from the node where Kibana is installed. The addresses that are external to the *management network* are not routable by default (see more details in step 7).

(c) Specify the LDAP server *Port* number or leave it empty to use the defaults.

(d) Specify the *Bind DN* of a user who has search privileges on the LDAP server.

(e) Specify the password of the user identified by the *Bind DN* selected in the above field.

(f) Specify the *User search base DN* in the Directory Information Tree (DIT) from where to search for users.

(g) Specify a valid attribute to search for users, for example, `uid`. The search should return a unique user entry.

(h) Specify a valid search filter to search for users, for example, `(objectClass=*)`

You can further restrict access to Kibana to those users who are members of a specific LDAP group:

(a) Select *Enable group-based authorization*.

(b) Specify the *LDAP attribute* in the user entry that identifies the LDAP group membership, for example, `memberUid`.

(c) Specify the DN of the LDAP group that will be mapped to the *admin role*.

(d) Specify the DN of the LDAP group that will be mapped to the *viewer role*.

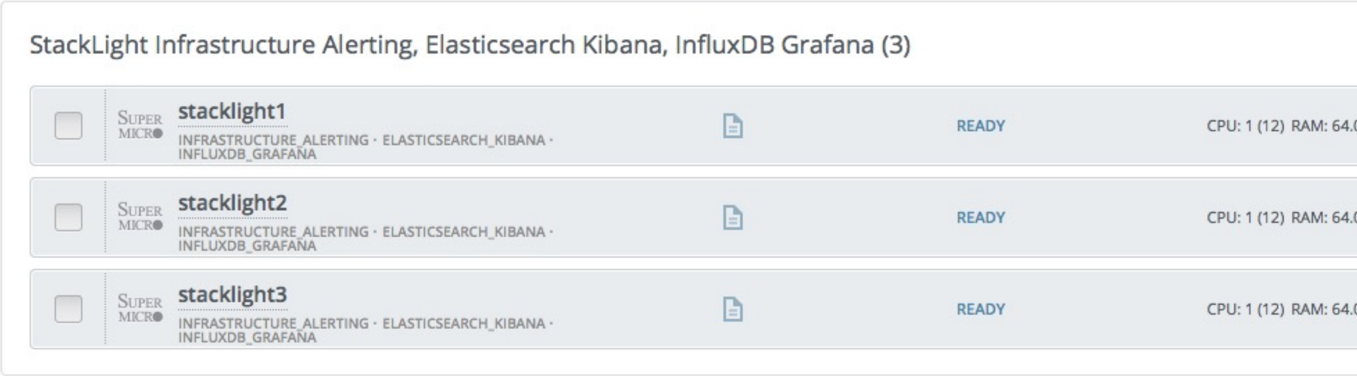
Users who have the *admin role* can modify the Kibana dashboards or create new ones. Users who have the *viewer role* can only view the Kibana dashboards.

7. In the Fuel web UI, configure your environment.

> **Caution:** By default, StackLight is configured to use the *management network* of the so-called Default Node Network Group. While this default setup may be appropriate for small deployments or evaluation purposes, we recommend not to use this network for StackLight in production. Instead, create a network dedicated to StackLight to improve performance and reduce the monitoring footprint. It will also facilitate access to the Kibana UI after deployment.

8. Click the *Nodes* tab and assign the *Elasticsearch_Kibana* role to the node(s) where you want to install the plugin.

   The example below shows that the *Elasticsearch_Kibana* role is assigned to three nodes alongside with the *Alerting_Infrastructure* and the *InfluxDB_Grafana* roles. The three plugins of the LMA toolchain back-end servers are installed on the same nodes.



StackLight Infrastructure Alerting, Elasticsearch Kibana, InfluxDB Grafana (3)

□	stacklight1 INFRASTRUCTURE_ALERTING · ELASTICSEARCH_KIBANA · INFLUXDB_GRAFANA		READY	CPU: 1 (12) RAM: 64.0
□	stacklight2 INFRASTRUCTURE_ALERTING · ELASTICSEARCH_KIBANA · INFLUXDB_GRAFANA		READY	CPU: 1 (12) RAM: 64.0
□	stacklight3 INFRASTRUCTURE_ALERTING · ELASTICSEARCH_KIBANA · INFLUXDB_GRAFANA		READY	CPU: 1 (12) RAM: 64.0

   Note: The Elasticsearch clustering for high availability requires that you assign the *Elasticsearch_Kibana* role to at least three nodes, but you can assign the *Elasticsearch_Kibana* role up to five nodes. You can also add or remove a node with the *Elasticsearch_Kibana* role after deployment.

9. If required, adjust the disk partitioning.

   By default, the Elasticsearch-Kibana plugin allocates:

   - 20% of the first available disk for the operating system by honoring a range of 15 GB minimum and 50 GB maximum.
   - 10 GB for `/var/log`.
   - At least 30 GB for the Elasticsearch database in `/opt/es-data`.

10. Deploy your environment.

2.3 Deploy an OpenStack environment using networking templates

By default, the Elasticsearch-Kibana cluster will be deployed on the Fuel management network. If this default configuration does not meet your requirements, you can leverage the Fuel networking templates' capability to change that default configuration and use a dedicated network instead.

Below is a networking template example to define a new network named `monitoring`. You can use this configuration example as a starting point and adapt it to your requirements.

```
adv_net_template:
  default:
    network_assignments:
      fuelweb_admin:
        ep: br-fw-admin
      management:
        ep: br-mgmt
      private:
        ep: br-mesh
      public:
        ep: br-ex
      storage:
        ep: br-storage
      monitoring:
        ep: br-monitoring
    network_scheme:
      admin:
        endpoints:
        - br-fw-admin
        roles:
          admin/pxe: br-fw-admin
          fw-admin: br-fw-admin
        transformations:
        - action: add-br
          name: br-fw-admin
        - action: add-port
          bridge: br-fw-admin
          name: <% if1 %>
      mgmt:
        endpoints:
        - br-mgmt
        roles:
          ceilometer/api: br-mgmt
          cinder/api: br-mgmt
          glance/api: br-mgmt
          heat/api: br-mgmt
          horizon: br-mgmt
          keystone/api: br-mgmt
          management: br-mgmt
          mgmt/api: br-mgmt
          mgmt/corosync: br-mgmt
          mgmt/database: br-mgmt
          mgmt/memcache: br-mgmt
          mgmt/messaging: br-mgmt
          mgmt/vip: br-mgmt
          mongo/db: br-mgmt
          murano/api: br-mgmt
          neutron/api: br-mgmt
          neutron/private: br-mgmt
          nova/api: br-mgmt
          nova/migration: br-mgmt
          rados_gw_management_vip: br-mgmt
          sahara/api: br-mgmt
          swift/api: br-mgmt
          swift/replication: br-mgmt
        transformations:
        - action: add-br
          name: br-mgmt
        - action: add-port
          bridge: br-mgmt
          name: <% if3 %>
      private:
        endpoints:
        - br-mesh
        roles:
          neutron/mesh: br-mesh
        transformations:
        - action: add-br
          name: br-mesh
        - action: add-port
          bridge: br-mesh
          name: <% if4 %>
      public:
        endpoints:
        - br-ex
        roles:
          ceph/radosgw: br-ex
          cinder/api: br-ex
          ex: br-ex
          neutron/floating: br-floating
          public/vip: br-ex
        transformations:
        - action: add-br
          name: br-ex
        - action: add-br
          name: br-floating
          provider: ovs
        - action: add-patch
          bridges:
          - br-floating
          - br-ex
          mtu: 65000
          provider: ovs
        - action: add-port
          bridge: br-ex
          name: <% if2 %>
      storage:
        endpoints:
        - br-storage
        roles:
          ceph/replication: br-storage
          cinder/iscsi: br-storage
          storage: br-storage
          swift/replication: br-storage
        transformations:
        - action: add-br
          name: br-storage
        - action: add-port
          bridge: br-storage
          name: <% if5 %>
      monitoring:
        endpoints:
        - br-monitoring
        roles:
          monitoring: br-monitoring
          elasticsearch: br-monitoring
          kibana: br-monitoring
          influxdb_vip: br-monitoring
          grafana: br-monitoring
          infrastructure_alerting: br-monitoring
          infrastructure_alerting_ui: br-monitoring
        transformations:
        - action: add-br
          name: br-monitoring
        - action: add-port
          bridge: br-monitoring
          name: <% if3 %>.101
    nic_mapping:
      default:
        # fw-admin
        if1: eth0
        # public
        if2: eth1
        # management + monitoring (VLAN: 101)
        if3: eth2
        # private
        if4: eth3
        # storage
        if5: eth4
    templates_for_node_role:
      # The following roles supports deployments using Neutron with tunneling␣
↪segmentation
      # and Cinder with LVM over iSCSI
      cinder:
      - admin
      - mgmt
      - private
      - storage
      - monitoring
      compute:
      - admin
      - mgmt
      - private
      - storage
      - monitoring
      controller:
      - admin
      - mgmt
      - public
      - private
      - storage
      - monitoring
      elasticsearch_kibana:
      - admin
      - mgmt
      - private
      - storage
      - monitoring
      influxdb_grafana:
      - admin
      - mgmt
      - private
      - storage
      - monitoring
      infrastructure_alerting:
      - admin
      - mgmt
      - private
      - storage
      - monitoring
```

To deploy an environment using networking templates:

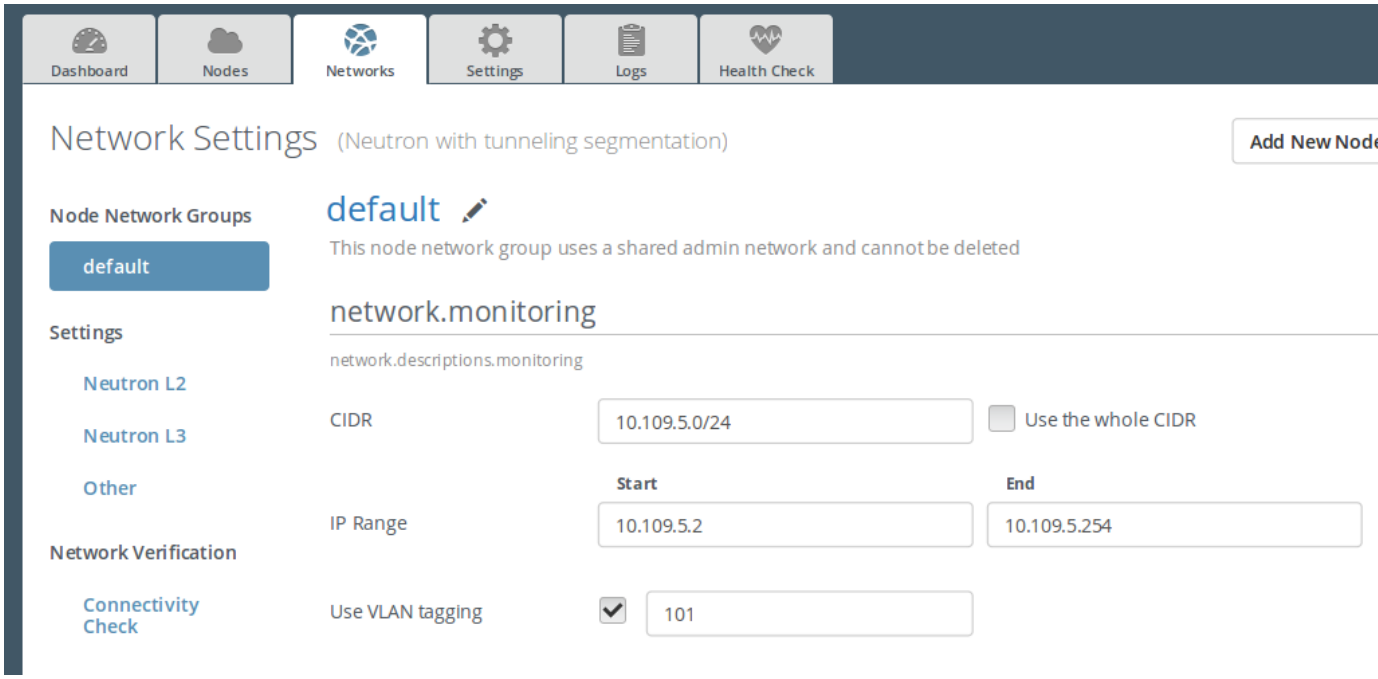
1. Upload the networking template:

```
$ fuel2 network-template upload -f ./network_template <ENVIRONMENT_ID>
```

2. Allocate an IP subnet for the `monitoring` network:

```
$ fuel2 network-group create -N <ENVIRONMENT_ID> -C 10.109.5.0/24 monitoring
```

3. Optional. Using the Fuel web UI, adjust the IP range:



Network Settings (Neutron with tunneling segmentation)

default

This node network group uses a shared admin network and cannot be deleted

network.monitoring

network.descriptions.monitoring

CIDR: 10.109.5.0/24 ☐ Use the whole CIDR

IP Range: Start 10.109.5.2 End 10.109.5.254

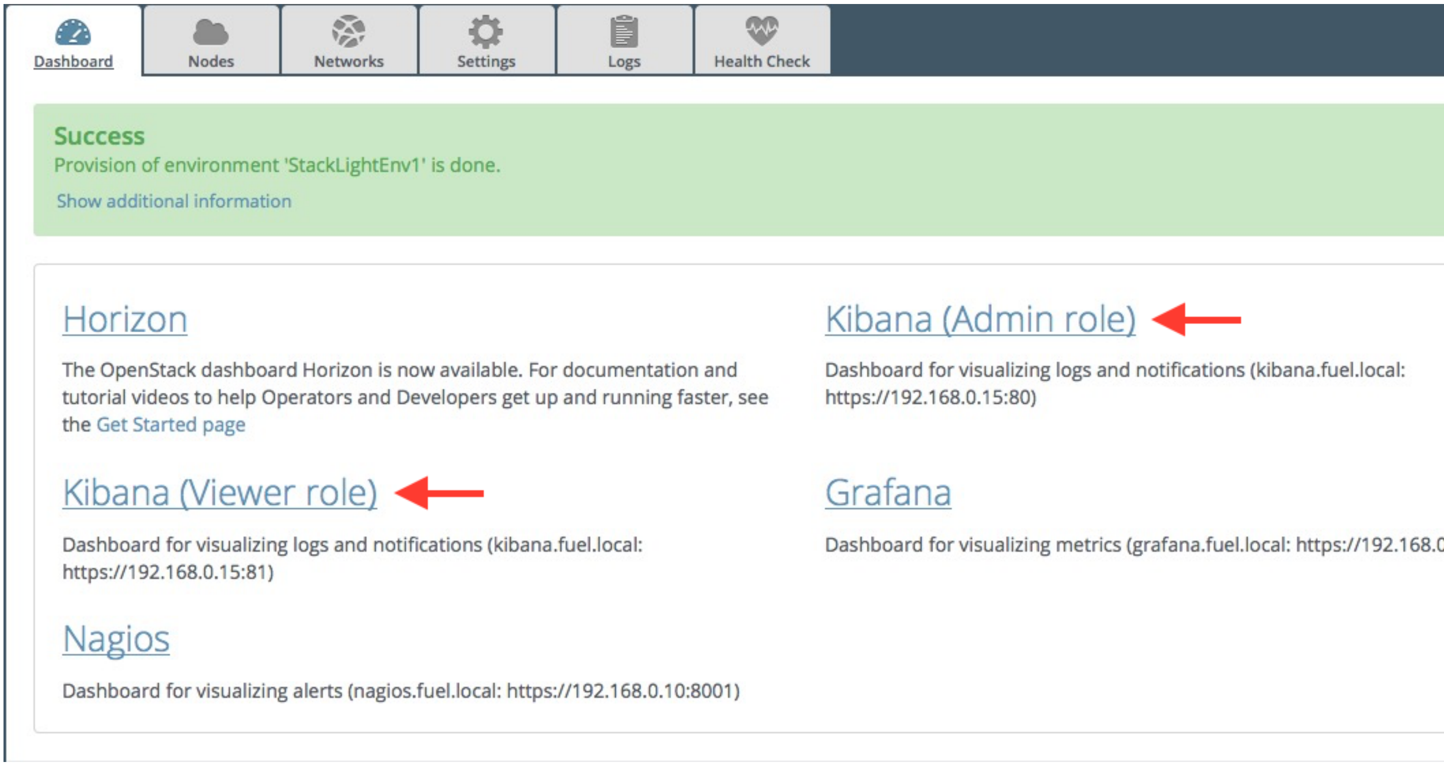
Use VLAN tagging ☑ 101

4. Proceed to *Configure the plugin*.

For details on networking templates, see the Fuel User Guide.

2.4 Verify the plugin after deployment

Depending on the number of nodes and deployment setup, deploying a Mirantis OpenStack environment can typically take from 20 minutes to several hours. But once your deployment is complete, you should see a deployment success notification message with two links to Kibana as shown in the picture below:



Success

Provision of environment 'StackLightEnv1' is done.

Show additional information

Horizon

The OpenStack dashboard Horizon is now available. For documentation and tutorial videos to help Operators and Developers get up and running faster, see the Get Started page

Kibana (Admin role)

Dashboard for visualizing logs and notifications (kibana.fuel.local: https://192.168.0.15:80)

Kibana (Viewer role)

Dashboard for visualizing logs and notifications (kibana.fuel.local: https://192.168.0.15:81)

Grafana

Dashboard for visualizing metrics (grafana.fuel.local: https://192.168.0

Nagios

Dashboard for visualizing alerts (nagios.fuel.local: https://192.168.0.10:8001)

Note: Two different ports are created to enforce the access authorization to Kibana:

- One port (80) for users with the *admin role*
- One port (81) for users with the *viewer role*.

If Kibana is installed on the *management network*, you may not have direct access to the Kibana web UI. Some extra network configuration may be required to create an SSH tunnel to the *management network*.

2.4.1 Verifying Elasticsearch

To verify that the Elasticsearch cluster is running properly, first retrieve the Elasticsearch cluster virtual IP address:

1. On the Fuel Master node, find the IP address of a node where the Elasticsearch server is installed using the **fuel nodes** command:

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

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

   ```
   root@node-1:~# hiera lma::elasticsearch::vip
   10.109.1.5
   ```

   Where `10.109.1.5` is the virtual IP address of your Elasticsearch cluster.
4. With that virtual IP address, run the following command:

   ```
   curl http://10.109.1.5:9200/
   ```

   The output should look as follows:

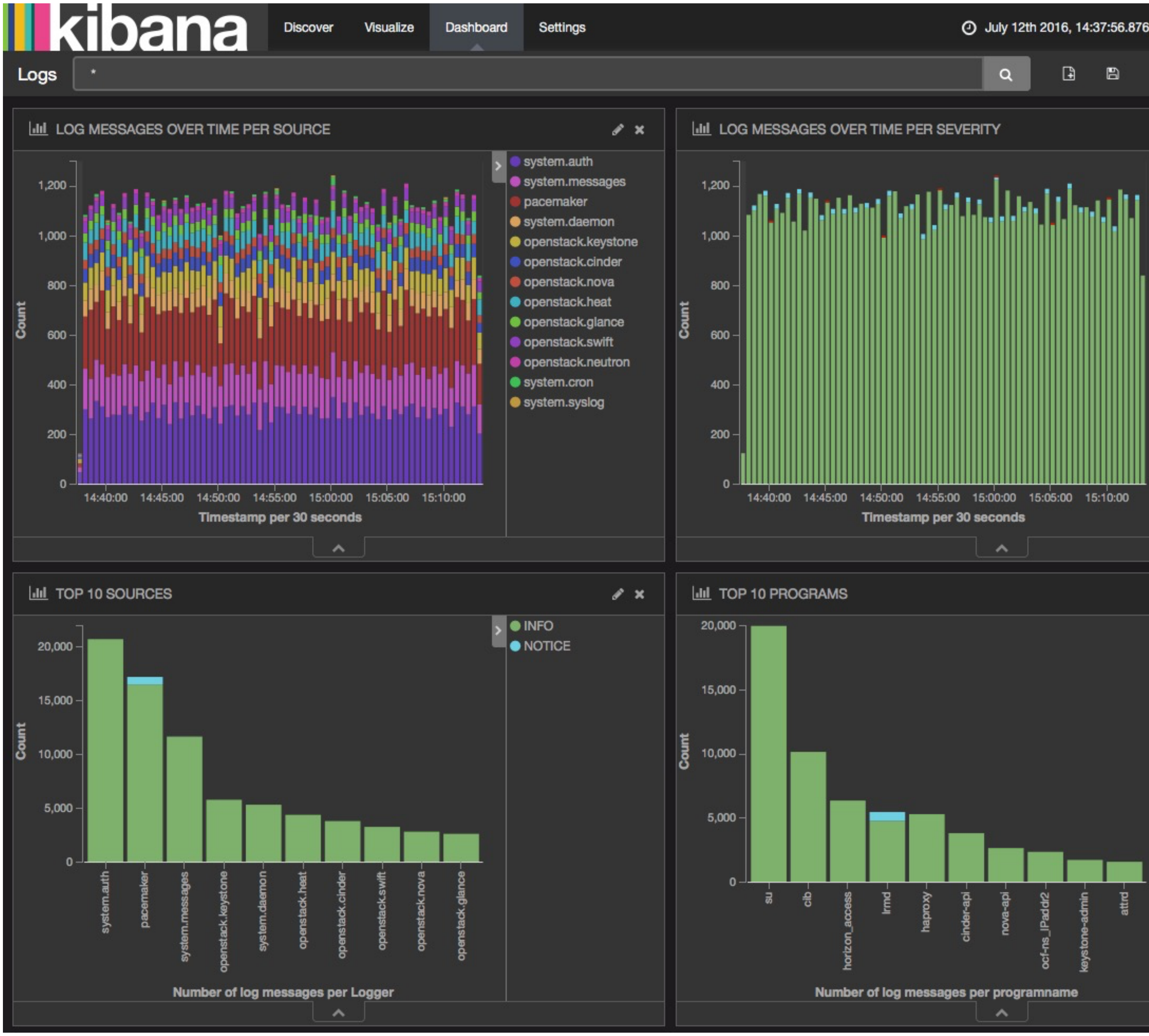
   ```
   {
     "status" : 200,
     "name" : "node-3.test.domain.local_es-01",
     "cluster_name" : "lma",
     "version" : {
       "number" : "1.7.4",
       "build_hash" : "0d3159b9fc8bc8e367c5c40c09c2a57c0032b32e",
       "build_timestamp" : "2015-12-15T11:25:18Z",
       "build_snapshot" : false,
       "lucene_version" : "4.10.4"
     },
     "tagline" : "You Know, for Search"
   }
   ```

2.4.2 Verifying Kibana

To verify the Kibana Dashboard:

1. Log in to the Fuel web UI.
2. Click on the *Kibana (Admin role)* link. If your DNS is not setup, enter the IP address and the port number.
3. Enter your credentials.

   You should be redirected to the Kibana **Logs Anaytics Dashboard** with four logs' sections as follows:



CHAPTER 3

Using the StackLight Elasticsearch-Kibana plugin for Fuel

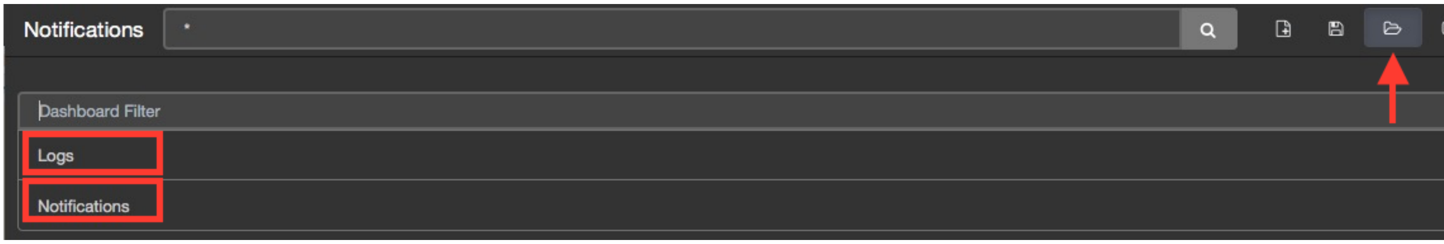
3.1 Use the plugin

3.1.1 Dashboards management

The StackLight Elasticsearch-Kibana plugin contains two built-in dashboards:

- The *Logs* Analytics Dashboard that is used to visualize and search the logs.
- The *Notifications* Analytics Dashboard that is used to visualize and search the OpenStack notifications if you enabled the feature in the Collector settings.

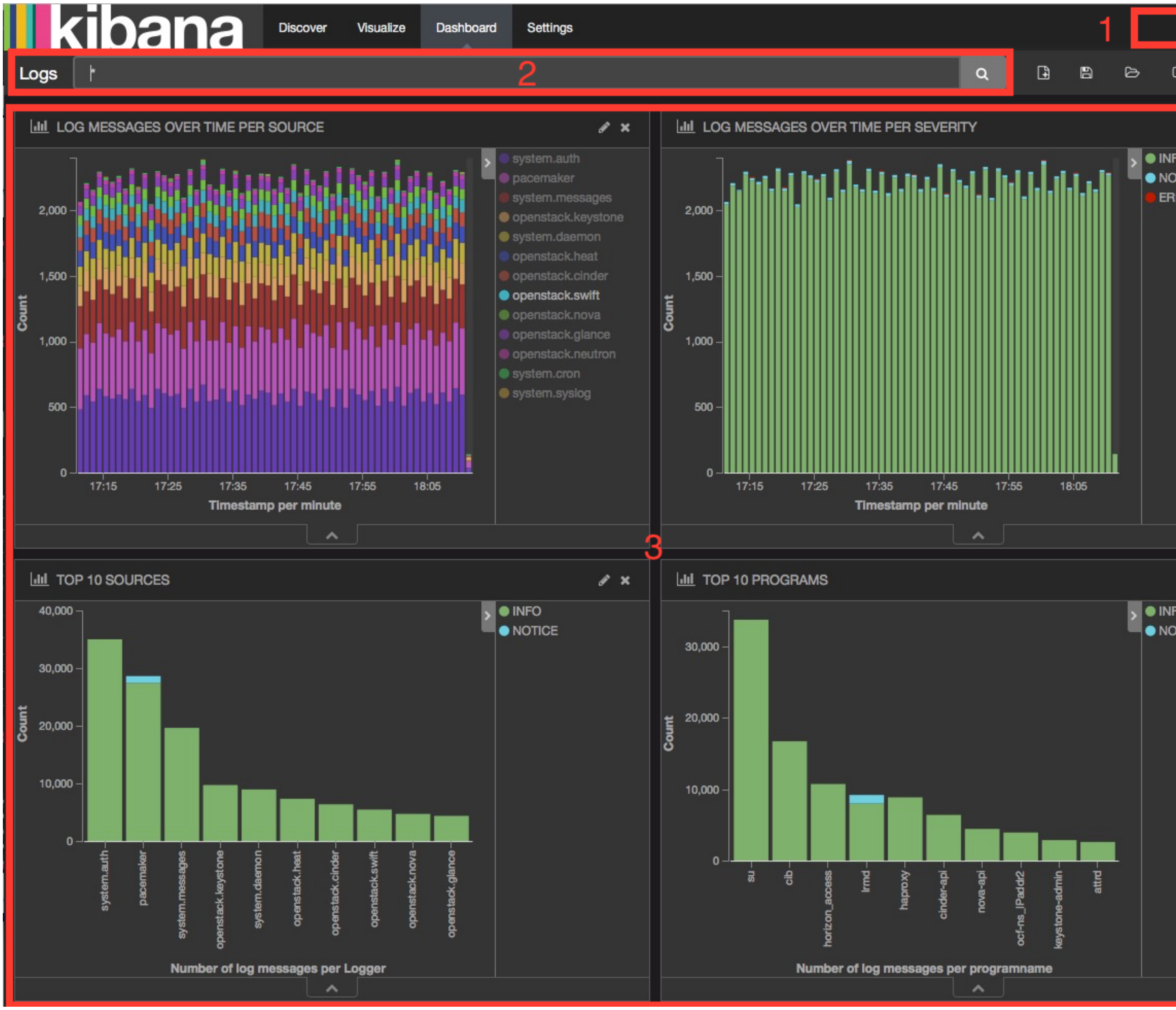
You can switch from one dashboard to another by clicking on the top-right *Load* icon on the toolbar to select the requested dashboard from the list, as shown below:



Each dashboard provides a single pane of glass for visualizing and searching all the logs and the notifications of your OpenStack environment.

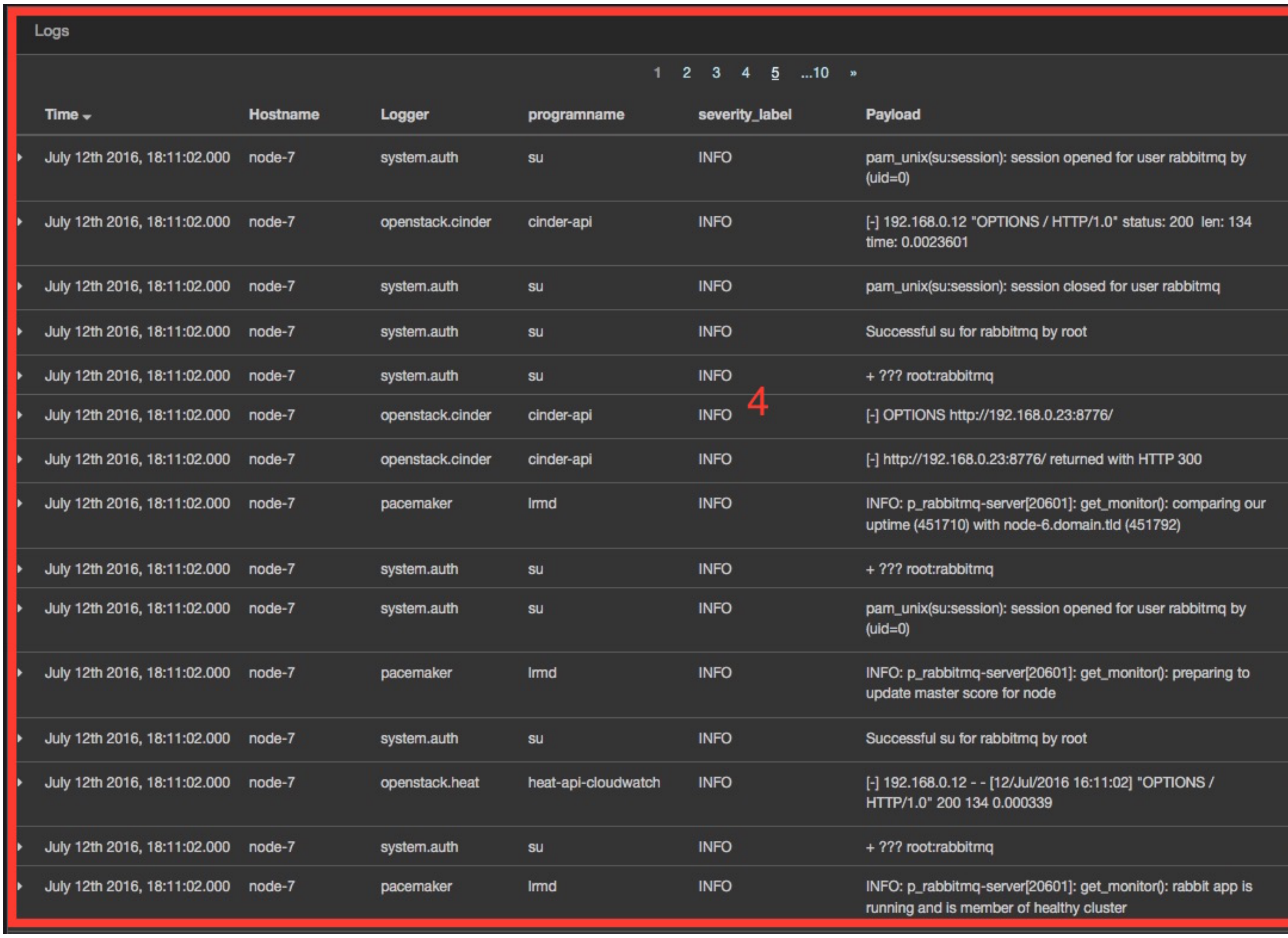
In the Collector settings, you can tag the logs by an environment name to distinguish which logs (and notifications) belong to what environment.

The Kibana Dashboard for logs is divided into several sections.



1. A time-picker control that lets you choose the time period you want to select and refresh frequency.
2. A text box to enter search queries.
3. Various logs analytics with six different panels:
   (a) A stack graph showing all the logs per source.
   (b) A stack graph showing all the logs per severity.
   (c) A stack graph showing all logs for top 10 sources.
   (d) A stack graph showing all the logs for top 10 programs.
   (e) A stack graph showing all logs for top 10 hosts.
   (f) A graph showing the number of logs per severity.
   (g) A graph showing the number of logs per role.

4. A table of log messages sorted in reverse chronological order.



Logs

1 2 3 4 5 ...10 »

Time	Hostname	Logger	programname	severity_label	Payload
July 12th 2016, 18:11:02.000	node-7	system.auth	su	INFO	pam_unix(su:session): session opened for user rabbitmq by (uid=0)
July 12th 2016, 18:11:02.000	node-7	openstack.cinder	cinder-api	INFO	[-] 192.168.0.12 "OPTIONS / HTTP/1.0" status: 200 len: 134 time: 0.0023601
July 12th 2016, 18:11:02.000	node-7	system.auth	su	INFO	pam_unix(su:session): session closed for user rabbitmq
July 12th 2016, 18:11:02.000	node-7	system.auth	su	INFO	Successful su for rabbitmq by root
July 12th 2016, 18:11:02.000	node-7	system.auth	su	INFO	+ ??? root:rabbitmq
July 12th 2016, 18:11:02.000	node-7	openstack.cinder	cinder-api	INFO	[-] OPTIONS http://192.168.0.23:8776/
July 12th 2016, 18:11:02.000	node-7	openstack.cinder	cinder-api	INFO	[-] http://192.168.0.23:8776/ returned with HTTP 300
July 12th 2016, 18:11:02.000	node-7	pacemaker	lrmd	INFO	INFO: p_rabbitmq-server[20601]: get_monitor(): comparing our uptime (451710) with node-6.domain.tld (451792)
July 12th 2016, 18:11:02.000	node-7	system.auth	su	INFO	+ ??? root:rabbitmq
July 12th 2016, 18:11:02.000	node-7	system.auth	su	INFO	pam_unix(su:session): session opened for user rabbitmq by (uid=0)
July 12th 2016, 18:11:02.000	node-7	pacemaker	lrmd	INFO	INFO: p_rabbitmq-server[20601]: get_monitor(): preparing to update master score for node
July 12th 2016, 18:11:02.000	node-7	system.auth	su	INFO	Successful su for rabbitmq by root
July 12th 2016, 18:11:02.000	node-7	openstack.heat	heat-api-cloudwatch	INFO	[-] 192.168.0.12 - - [12/Jul/2016 16:11:02] "OPTIONS / HTTP/1.0" 200 134 0.000339
July 12th 2016, 18:11:02.000	node-7	system.auth	su	INFO	+ ??? root:rabbitmq
July 12th 2016, 18:11:02.000	node-7	pacemaker	lrmd	INFO	INFO: p_rabbitmq-server[20601]: get_monitor(): rabbit app is running and is member of healthy cluster

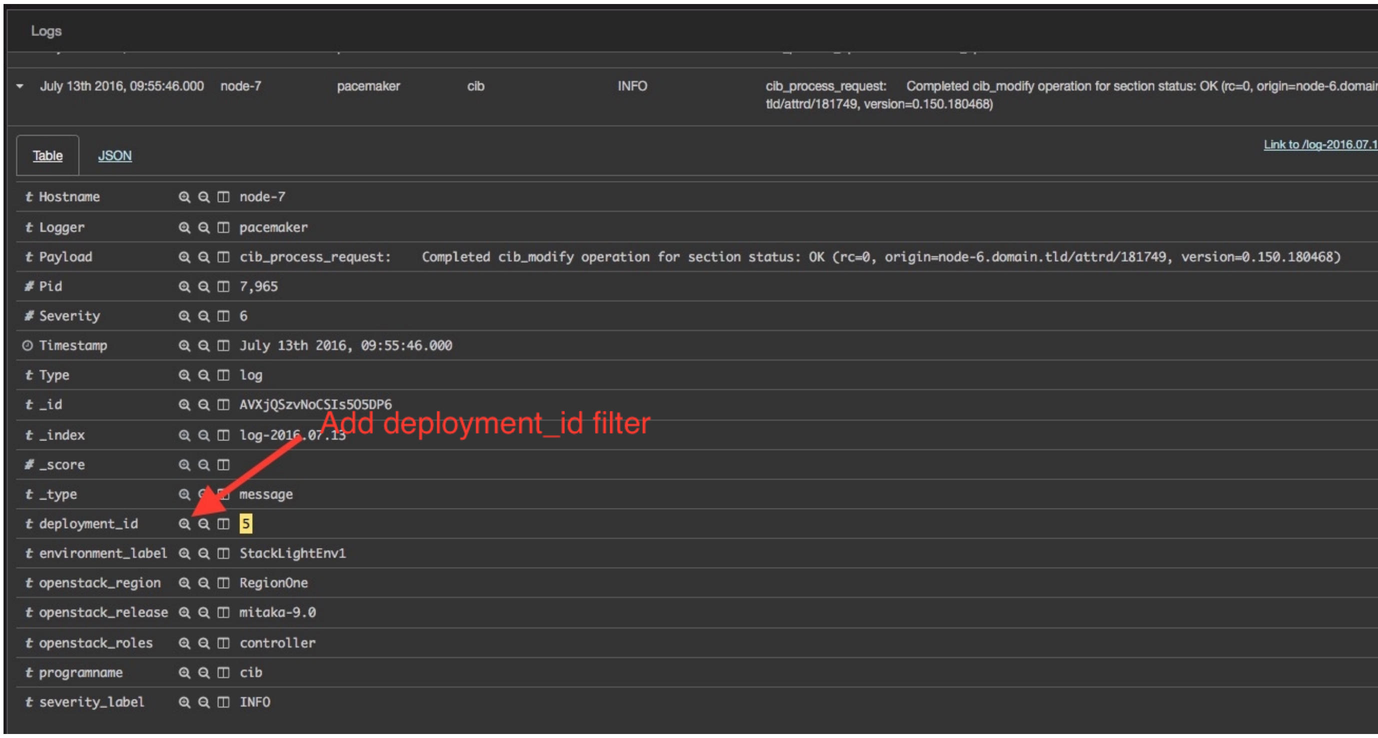
3.1.2 Filters and queries

Filters and queries have similar syntax but they are used for different purposes:

- The filters are used to restrict what is displayed in the Dashboard.
- The queries are used for free-text search.

You can combine multiple queries and compare their results. You can also further filter the log messages. For example, to select *deployment_id*:

1. Expand a log entry.
2. Select the *deployment_id* field by clicking on the magnifying glass icon as shown below:



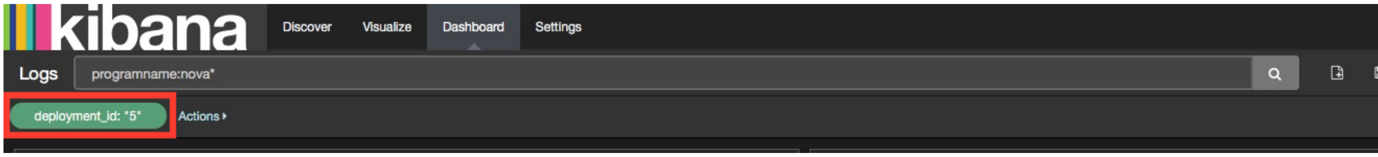
Logs

July 13th 2016, 09:55:46.000 node-7 pacemaker cib INFO cib_process_request: Completed cib_modify operation for section status: OK (rc=0, origin=node-6.domai tld/attrd/181749, version=0.150.180468)

Table JSON Link to /log-2016.07.1

t Hostname	node-7
t Logger	pacemaker
t Payload	cib_process_request: Completed cib_modify operation for section status: OK (rc=0, origin=node-6.domain.tld/attrd/181749, version=0.150.180468)
# Pid	7,965
# Severity	6
Timestamp	July 13th 2016, 09:55:46.000
t Type	log
t _id	AVXjQSzvNoCSIs5O5DP6
t _index	log-2016.07.13
# _score	
t _type	message
t deployment_id	5
t environment_label	StackLightEnv1
t openstack_region	RegionOne
t openstack_release	mitaka-9.0
t openstack_roles	controller
t programname	cib
t severity_label	INFO

This will apply a new filter in the Dashboard:



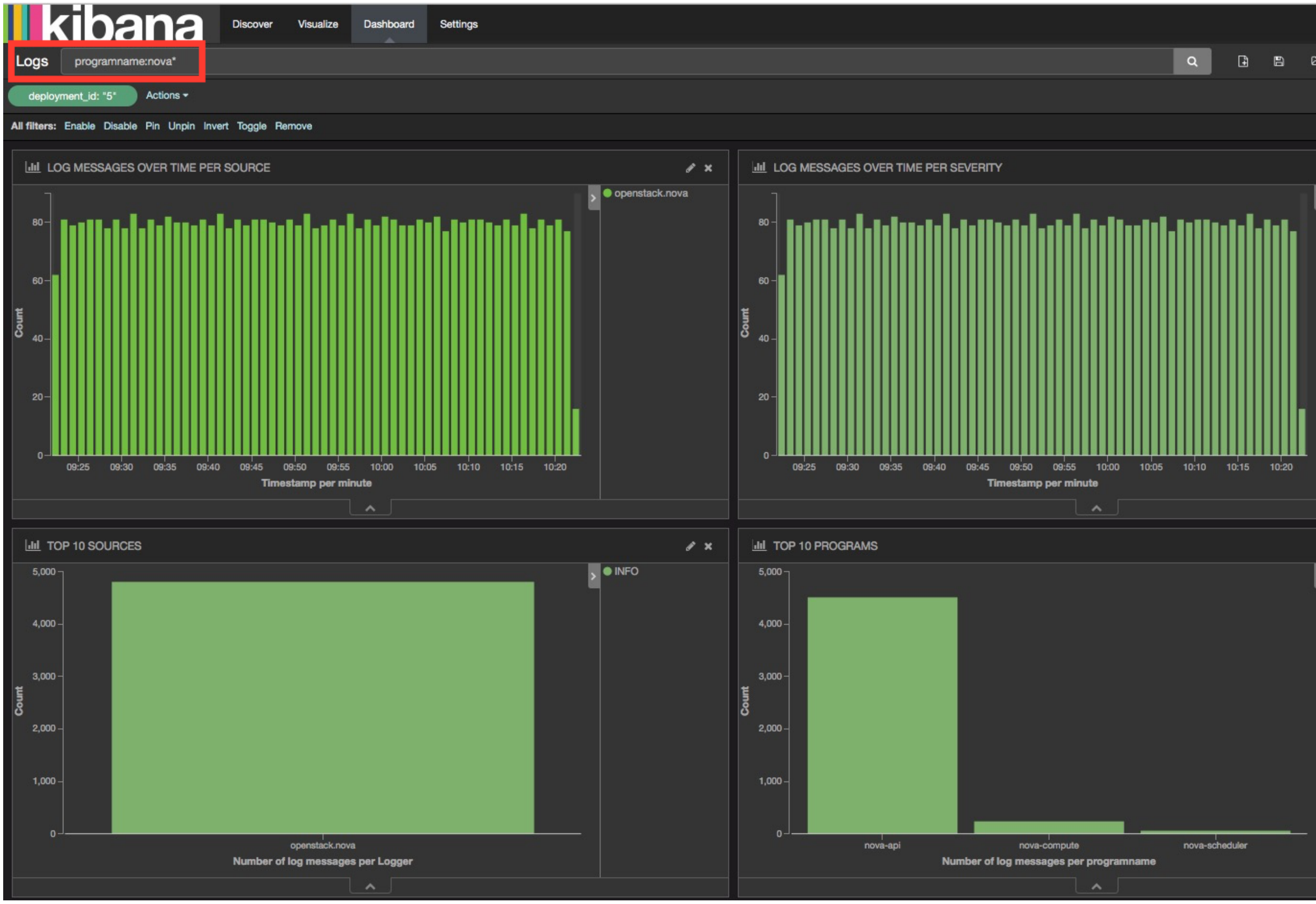
programname:nova*

deployment_id: "5"

Filtering works for any field that has been indexed for the log entries that are in the Dashboard.

Filters and queries can also use wildcards that can be combined with the *field names* like in `programname: <name>*`.

For example, to display only the Nova logs, enter `programname:nova*` in the query text box as shown below:



Logs programname:nova*

deployment_id: "5"

3.2 Troubleshooting

If you cannot access the Kibana Dashboard or you get no data in the Dashboard, use the following troubleshooting tips:

1. Verify that the StackLight Collector is running properly. For details, see the StackLight Collector troubleshooting instructions.
2. Verify that the nodes can connect to the Elasticsearch cluster through the virtual IP address on port `9200` as described in the *Verifying Elasticsearch* section.
3. On any of the *Elasticsearch_Kibana* role nodes, check the status of the virtual IP address and HAProxy resources on the Pacemaker cluster:

```
root@node-1:~# crm resource status vip__es_vip_mgmt
resource vip__es_vip_mgmt is running on: node-1.test.domain.local

root@node-1:~# crm resource status p_haproxy
resource p_haproxy is running on: node-1.test.domain.local
```

4. If the virtual IP or HAProxy resources are down, restart them:

```
root@node-1:~# crm resource start vip__es_vip_mgmt
root@node-1:~# crm resource start p_haproxy
```

5. Verify that the Elasticsearch server is up and running on both CentOS and Ubuntu:

```
[root@node-1 ~]# /etc/init.d/elasticsearch-es-01 status
```

If Elasticsearch is down, restart it on both CentOS and Ubuntu:

```
[root@node-1 ~]# /etc/init.d/elasticsearch-es-01 start
```

6. Verify that Apache is up and running on both CentOS and Ubuntu:

```
[root@node-1 ~]# /etc/init.d/apache2 status
```

If Apache is down, restart it on both CentOS and Ubuntu:

```
[root@node-1 ~]# /etc/init.d/apache2 start
```

7. Look for errors in the Elasticsearch log files located at `/var/log/elasticsearch/es-01/`.
8. Look for errors in the Apache log files located at `/var/log/apache2/`.

3.3 Advanced operations

This section describes advanced operations that you can apply to your Elasticsearch cluster using the StackLight Elasticsearch-Kibana Fuel plugin.

3.3.1 Cluster operations

Because of limitations in the current implementation of the plugin, manual operations are required after the Elasticsearch cluster is scaled up or scaled down. Using these operations, you can adjust the replication factor of the Elasticsearch indices that are based on the new number of nodes on the cluster.

The plugin uses three types of indices:

- The log indices named *log-%{+YYYY.MM.dd}* which are created on a daily basis.
- The notification indices named *notification-%{+YYYY.MM.dd}* which are created on a daily basis.
- The Kibana index named *kibana-int* which is created once during the installation to store the templates of the Kibana dashboards.

Adjusting the replication factor for the *kibana-int* index is performed automatically by the plugin. Therefore, no manual operation is required for this index when the cluster is scaled up or down. But this is not the case for the replication factor of other two indices that you should manually update as described in the Elasticsearch official documentation.

The following sections describe in detail how to scale up and scale down the Elasticsearch cluster. Scaling up from one node to three nodes and scaling down from three nodes to one node are used as examples. Your mileage may vary, but the principal of (re)configuring the replication factor of the indices should remain the same.

Scaling up

The problem that the manual operation aims to address is that the replication factor for the old indices is not updated automatically by the plugin when a new node is added in the cluster. If you want the old indices to be replicated on the new node(s), adjust the *number_of_replicas* parameter to the current size of the cluster for those indices as shown below.

The output below shows that the replication factor of the indices created before the scale-up is zero. In this example, a scale-up was performed on the 3rd of February. Therefore, the indices created after that date (*log-2016.02.04* in this example) are automatically updated with the correct number of replicas (number of cluster nodes - 1).

```
[root@node-1 ~]# curl <VIP>:9200/_cat/indices?v
health status index                      pri rep docs.count docs.deleted ....
green  open   log-2016.02.03               5   0     270405            0 ....
green  open   log-2016.02.04               5   2    1934581            0 ....
```

If you want the *log-2016.02.03* index to be replicated, update the *number_of_replicas* parameter of that index:

```
[root@node-1 ~]# curl -XPUT  <VIP>:9200/log-2016.02.03/_settings \
  -d '{ "index": {"number_of_replicas": 2}}'
{"acknowledged":true}

[root@node-1 ~]# curl <VIP>:9200/_cat/indices?v
health status index                      pri rep docs.count docs.deleted ....
green  open   log-2016.02.03               5   2     270405            0 ....
green  open   log-2016.02.04               5   2    1934581            0 ....
```

> **Caution:** Replicating the old indices on the new node(s) will increase the load on the cluster as well as the size required to store the data.

Scaling down

After a scale-down, align the *number_of_replicas* of all indices with the new size of the cluster. Otherwise, StackLight reports a critical status of the Elasticsearch cluster:

```
[root@node-1 ~]# # the current index health is 'red' after the scale-down
[root@node-1 ~]# curl <VIP>:9200/_cat/indices?v
health  status index                  pri rep docs.count ....
red     open   log-2016.02.04           5   2    1934581 ....

[root@node-1 ~]# curl -XPUT  <VIP>:9200/log-2016.02.04/_settings \
  -d '{"index": {"number_of_replicas": 0}}'
{"acknowledged":true}

[root@node-1 ~]# # the cluster health is now 'green'
[root@node-1 ~]# curl <VIP>:9200/_cat/indices?v
health  status index                  pri rep docs.count ....
green   open   log-2016.02.04           5   2    1934581 ....
```