# The LMA Collector Plugin for Fuel Documentation for QA

***Release 0.8.0***

**Mirantis Inc.**

May 11, 2016

Contents

CHAPTER 1

# QA documentation

## 1.1 Test Strategy

The test plan implements system, functional and non-functional tests. These tests will be automated but tests of the user interfaces will have to be done manually.

### 1.1.1 Acceptance Criteria

1. The plugins can be installed and enabled on the Fuel master node.
2. The LMA Collector service is deployed on all the nodes of the environment including nodes with the 'base-os' role and custom roles (influxdb_grafana, elasticsearch_kibana, infrastructure_alerting).
3. The Elasticsearch server and the Kibana UI are deployed on one node with the elasticsearch_kibana role.
4. The InfluxDB server and the Grafana UI are deployed on one node with the influxdb_grafana role.
5. The Nagios server and dashboard are deployed on one node with the infrastructure_alerting role.
6. Kibana UI can be used to index and search both log messages and notifications.
7. The Grafana dashboards display detailed metrics for the main OpenStack services.
8. The Nagios UI displays status of all nodes and OpenStack services.
9. The plugins can be uninstalled when no environment uses them.

### 1.1.2 Test environment, infrastructure and tools

The 4 LMA plugins are installed on the Fuel master node.

For the controller nodes, it is recommended to deploy on hosts with at least 2 CPUs and 4G of RAM.

### 1.1.3 Product compatibility matrix

| Product | Version/Comment |
|---|---|
| Mirantis OpenStack | 7.0 |
| LMA collector plugin | 0.8.0 |
| Elasticsearch-Kibana plugin | 0.8.0 |
| InfluxDB-Grafana plugin | 0.8.0 |
| LMA Infrastructure Alerting plugin | 0.8.0 |

## 1.2 System testing

### 1.2.1 Install the plugins

| Test Case ID | install_lma_plugins |
|---|---|
| Description | Verify that the plugins can be installed. |
| Prerequisites | N/A |

#### Steps

1. Copy the 4 plugins to the Fuel master node using scp.
2. Connect to the Fuel master node using ssh.
3. Install the plugins using the fuel CLI.
4. Connect to the Fuel web UI.
5. Create a new environment using the Fuel UI Wizard.
6. Click on the Plugins tab.

#### Expected Result

The 4 plugins are present in the Fuel UI.

### 1.2.2 Deploy an environment with the plugins

| Test Case ID | deploy_lma_plugins |
|---|---|
| Description | Verify that the plugins can be deployed. |
| Prerequisites | Plugins are installed on the Fuel master node (see *Install the plugins*). |

#### Steps

1. Connect to the Fuel web UI.
2. Create a new environment with the Fuel UI wizard with the default settings.
3. Click on the Settings tab of the Fuel web UI.
4. Select the LMA collector plugin tab and fill-in the following fields:
   (a) Enable the plugin.
   (b) Select 'Local node' for "Event analytics".
   (c) Select 'Local node' for "Metric analytics".
   (d) Select 'Alerts sent to a local node running the LMA Infrastructure Alerting plugin' for "Alerting".
5. Select the Elasticsearch-Kibana plugin tab and enable it.
6. Select the InfluxDB-Grafana plugin and fill-in the required fields:
   (a) Enable the plugin.
   (b) Enter 'lmapass' as the root, user and grafana user passwords.

7. Select the LMA Infrastructure-Alerting plugin and fill-in the required fields:
   (a) Enable the plugin.
   (b) Enter 'root@localhost' as the recipient
   (c) Enter 'nagios@localhost' as the sender
   (d) Enter '127.0.0.1' as the SMTP server address
   (e) Choose "None" for SMTP authentication (default)
8. Click on the Nodes tab of the Fuel web UI.
9. Assign roles to nodes:
   (a) 1 node with these 3 roles (this node is referenced later as the 'lma' node):
      i. influxdb_grafana
      ii. elasticsearch_kibana
      iii. infrastructure_alerting
   (b) 3 nodes with the 'controller' role
   (c) 1 node with the 'compute' + 'cinder' node
10. Click 'Deploy changes'.
11. Once the deployment has finished, connect to each node of the environment using ssh and run the following checks:
   (a) Check that hekad and collectd processes are up and running on all the nodes as described in the LMA Collector documentation.
   (b) Look for errors in /var/log/lma_collector.log
   (c) Check that the node can connect to the Elasticsearch server (`http://<IP address of the 'lma' node>:9200/`)
   (d) Check that the node can connect to the InfluxDB server (`http://<IP address of the 'lma' node>:8086/`)
12. Check that the dashboards are running
   (a) Check that you can connect to the Kibana UI (`http://<IP address of the 'lma' node>:80/`)
   (b) Check that you can connect to the Grafana UI (`http://<IP address of the 'lma' node>:8000/`) with user='lma', password='lmapass'
   (c) Check that you can connect to the Nagios UI (`http://<IP address of the 'lma' node>:8001/`) with user='nagiosadmin', password='r00tme'

#### Expected Result

The environment is deployed successfully.

### 1.2.3 Add/remove controller nodes in existing environment

| | |
|---|---|
| Test Case ID | modify_env_with_plugin_remove_add_controller |
| Description | Verify that the number of controllers can scale up and down. |
| Prerequisites | Environment deployed with the 4 plugins (See *Deploy an environment with the plugins*). |

#### Steps

1. Remove 1 node with the controller role.
2. Re-deploy the cluster.
3. Check the plugin services using the CLI
4. Check in the Nagios UI that the removed node is no longer monitored.
5. Run the health checks (OSTF).
6. Add 1 new node with the controller role.
7. Re-deploy the cluster.
8. Check the plugin services using the CLI.
9. Check in the Nagios UI that the new node is monitored.
10. Run the health checks (OSTF).

#### Expected Result

The OSTF tests pass successfully.

All the plugin services are running and work as expected after each modification of the environment.

The Nagios service has been reconfigured to take care of the node removal and addition.

### 1.2.4 Add/remove compute nodes in existing environment

| | |
|---|---|
| Test Case ID | modify_env_with_plugin_remove_add_compute |
| Description | Verify that the number of computes can scale up and down. |
| Prerequisites | Environment deployed with the 4 plugins (See *Deploy an environment with the plugins*). |

#### Steps

1. Remove 1 node with the compute role.
2. Re-deploy the cluster.
3. Check the plugin services using the CLI
4. Check in the Nagios UI that the removed node is no longer monitored.
5. Run the health checks (OSTF).
6. Add 1 new node with the compute role.
7. Re-deploy the cluster.
8. Check the plugin services using the CLI.
9. Check in the Nagios UI that the new node is monitored.
10. Run the health checks (OSTF).

#### Expected Result

The OSTF tests pass successfully.

All the plugin services are running and work as expected after each modification of the environment.

The Nagios service has been reconfigured to take care of the node removal and addition.

### 1.2.5 Uninstall the plugins with deployed environment

| | |
|---|---|
| Test Case ID | uninstall_plugin_with_deployed_env |
| Description | Verify that the plugins can be uninstalled after the deployed environment is removed. |
| Prerequisites | Environment deployed with the 4 plugins (see *Deploy an environment with the plugins*). |

#### Steps

1. Try to remove the plugins using the Fuel CLI and ensure that the command fails with "Can't delete plugin which is enabled for some environment".
2. Remove the environment.
3. Remove the plugins.

#### Expected Result

An alert is raised when we try to delete plugins which are attached to an active environment.

After the environment is removed, the plugins are removed successfully too.

### 1.2.6 Uninstall the plugins

| | |
|---|---|
| Test Case ID | uninstall_plugin |
| Description | Verify that the plugins can be uninstalled. |
| Prerequisites | The 4 plugins are installed on the Fuel node (see *Install the plugins*). |

#### Steps

1. Remove the plugins.

#### Expected Result

The plugins are removed.

## 1.3 Functional testing

### 1.3.1 Display and query logs in the Kibana UI

| | |
|---|---|
| Test Case ID | query_logs_in_kibana_ui |
| Description | Verify that the logs show up in the Kibana UI. |
| Prerequisites | Environment deployed with the 4 plugins (see *Deploy an environment with the plugins*). |

#### Steps

1. Open the Kibana URL at `http://<IP address of the 'lma' node>/`
2. Enter 'programname:nova*' in the Query box.
3. Check that Nova logs are displayed.

#### Expected Result

The Kibana UI displays entries for all the controller and compute nodes deployed in the environment.

### 1.3.2 Display and query Nova notifications in the Kibana UI

| | |
|---|---|
| Test Case ID | query_nova_notifications_in_kibana_ui |
| Description | Verify that the Nova notifications show up in the Kibana UI. |
| Prerequisites | Environment deployed with the 4 plugins (see *Deploy an environment with the plugins*). |

#### Steps

1. Launch, update, rebuild, resize, power-off, power-on, snapshot, suspend, shutdown, and delete an instance in the OpenStack environment (using the Horizon dashboard for example) and write down the instance's id.
2. Open the Kibana URL at `http://<IP address of the 'lma' node>/`
3. Open the Notifications dashboard using the 'Load' icon.
4. Enter 'instance_id:<uuid>' in the Query box where <uuid> is the id of the launched instance.

#### Expected Result

All event types for Nova are listed except compute.instance.create.error and compute.instance.resize.revert.{start|end}.

### 1.3.3 Display and query Glance notifications in the Kibana UI

| | |
|---|---|
| Test Case ID | query_glance_notifications_in_kibana_ui |
| Description | Verify that the Glance notifications show up in the Kibana UI. |
| Prerequisites | Environment deployed with the 4 plugins (see *Deploy an environment with the plugins*). |

#### Steps

1. Run the OSTF platform test "Check create, update and delete image actions using Glance v2".
2. Open the Kibana URL at `http://<IP address of the 'lma' node>/`
3. Open the Notifications dashboard using the 'Load' icon.
4. Enter 'glance' in the Query box.

#### Expected Result

All event types for Glance are listed.

### 1.3.4 Display and query Cinder notifications in the Kibana UI

| Test Case ID | query_cinder_notifications_in_kibana_ui |
|---|---|
| Description | Verify that the cinder notifications show up in the Kibana UI. |
| Prerequisites | Environment deployed with the 4 plugins (see *Deploy an environment with the plugins*). |

#### Steps

1. Create and update a volume in the OpenStack environment (using the Horizon dashboard for example) and write down the volume id.
2. Open the Kibana URL at `http://<IP address of the 'lma' node>/`
3. Open the Notifications dashboard using the 'Load' icon.
4. Enter 'volume_id:<uuid>' in the Query box where <uuid> is the id of the created volume.

#### Expected Result

All event types for Cinder are listed.

### 1.3.5 Display and query Heat notifications in the Kibana UI

| Test Case ID | query_heat_notifications_in_kibana_ui |
|---|---|
| Description | Verify that the heat notifications show up in the Kibana UI. |
| Prerequisites | Environment deployed with the 4 plugins (see *Deploy an environment with the plugins*). |

#### Steps

1. Run all OSTF Heat platform tests.
2. Open the Kibana URL at `http://<IP address of the 'lma' node>/`
3. Open the Notifications dashboard using the 'Load' icon.
4. Enter 'heat' in the Query box.

#### Expected Result

All event types for Heat are listed.

### 1.3.6 Display and query Neutron notifications in the Kibana UI

| | |
|---|---|
| Test Case ID | query_neutron_notifications_in_kibana_ui |
| Description | Verify that the Neutron notifications show up in the Kibana UI. |
| Prerequisites | Environment deployed with the 4 plugins (see *Deploy an environment with the plugins*). |

#### Steps

1. Run OSTF functional tests: 'Create security group' and 'Check network connectivity from instance via floating IP'.
2. Open the Kibana URL at `http://<IP address of the 'lma' node>/`
3. Open the Notifications dashboard using the 'Load' icon.
4. Enter 'neutron' in the Query box.

#### Expected Result

All event types for Neutron are listed.

### 1.3.7 Display and query Keystone notifications in the Kibana UI

| | |
|---|---|
| Test Case ID | query_keystone_notifications_in_kibana_ui |
| Description | Verify that the Keystone notifications show up in the Kibana UI. |
| Prerequisites | Environment deployed with the 4 plugins (see *Deploy an environment with the plugins*). |

#### Steps

1. Run OSTF platform test: 'Create user and authenticate with it to Horizon'.
2. Open the Kibana URL at `http://<IP address of the 'lma' node>/`
3. Open the Notifications dashboard using the 'Load' icon.
4. Enter 'keystone' in the Query box.

#### Expected Result

All event types for Keystone are listed.

### 1.3.8 Display the dashboards in the Grafana UI

| | |
|---|---|
| Test Case ID | display_dashboards_in_grafana_ui |
| Description | Verify that the dashboards show up in the Grafana UI. |
| Prerequisites | Environment deployed with the 4 plugins (see *Deploy an environment with the plugins*). |

#### Steps

1. Open the Grafana URL at `http://<IP address of the 'lma' node>:8000/`
2. Sign-in using the credentials provided during the configuration of the environment.
3. Go to the Main dashboard and verify that everything is ok.
4. Repeat the previous step for the following dashboards:
   (a) Cinder
   (b) Glance
   (c) Heat
   (d) Keystone
   (e) Nova
   (f) Neutron
   (g) HAProxy
   (h) RabbitMQ
   (i) MySQL
   (j) Apache
   (k) Memcached
   (l) System
   (m) LMA Self-monitoring

#### Expected Result

The Grafana UI shows the overall status of the OpenStack services and detailed statistics about the selected controller.

### 1.3.9 Display the Nova metrics in the Grafana UI

| | |
|---|---|
| Test Case ID | display_nova_metrics_in_grafana_ui |
| Description | Verify that the Nova metrics show up in the Grafana UI. |
| Prerequisites | Environment deployed with the 4 plugins (see *Deploy an environment with the plugins*). |

#### Steps

1. Open the Grafana URL at `http://<IP address of the 'lma' node>:8000/`
2. Sign-in using the credentials provided during the configuration of the environment.
3. Go to the Nova dashboard.
4. Connect to the Fuel web UI, launch the full suite of OSTF tests and wait for their completion.
5. Check that the 'instance creation time' graph in the Nova dashboard reports values.

#### Expected Result

The Grafana UI shows the instance creation time over time.

### 1.3.10 Report service alerts with warning severity

| | |
|---|---|
| Test Case ID | report_service_alerts_with_warning_severity |
| Description | Verify that the warning alerts for services show up in the Grafana and Nagios UI. |
| Prerequisites | Environment deployed with the 4 plugins (see *Deploy an environment with the plugins*). |

#### Steps

1. Open the Grafana URL at `http://<IP address of the 'lma' node>:8000/` and load the Nova dashboard.
2. Open the Nagios URL at `http://<IP address of the 'lma' node>:8001/` in another tab and click the 'Services' menu item.
3. Connect to one of the controller nodes using ssh and stop the nova-api service.
4. Wait for at least 1 minute.
5. On Grafana, check the following items:
   (a) the box in the upper left corner of the dashboard displays 'WARN' with an orange background,
   (b) the API panels report 1 entity as down.
6. On Nagios, check the following items:
   (a) the 'nova' service is in 'WARNING' state,
   (b) the local user root on the lma node has received an email about the service being in warning state.
7. Restart the nova-api service.
8. Wait for at least 1 minute.
9. On Grafana, check the following items:
   (a) the box in the upper left corner of the dashboard displays 'OKAY' with an green background,
   (b) the API panels report 0 entity as down.
10. On Nagios, check the following items:
    (a) the 'nova' service is in 'OK' state,
    (b) the local user root on the lma node has received an email about the recovery of the service.
11. Stop the nova-scheduler service.
12. Wait for at least 3 minutes.
13. On Grafana, check the following items:
    (a) the box in the upper left corner of the dashboard displays 'WARN' with an orange background,
    (b) the scheduler panel reports 1 entity as down.
14. On Nagios, check the following items:
    (a) the 'nova' service is in 'WARNING' state,
    (b) the local user root on the lma node has received an email about the service being in warning state.
15. Restart the nova-scheduler service.
16. Wait for at least 1 minute.
17. On Grafana, check the following items:

(a) the box in the upper left corner of the dashboard displays 'OKAY' with an green background,

(b) the scheduler panel reports 0 entity as down.

18. On Nagios, check the following items:
    (a) the 'nova' service is in 'OK' state,
    (b) the local user root on the lma node has received an email about the recovery of the service.
19. Repeat steps 2 to 18 for the following services:
    (a) Cinder (stopping and starting the cinder-api and cinder-scheduler services respectively).
    (b) Neutron (stopping and starting the neutron-server and neutron-openvswitch-agent services respectively).
20. Repeat steps 2 to 10 for the following services:
    (a) Glance (stopping and starting the glance-api service).
    (b) Heat (stopping and starting the heat-api service).
    (c) Keystone (stopping and starting the Apache service).

#### Expected Result

The Grafana UI shows that the global service status goes from ok to warning and back to ok. It also reports detailed information about which entity is missing.

The Nagios UI shows that the service status goes from ok to warning and back to ok. Alerts are sent by email to the configured recipient.

### 1.3.11 Report service alerts with critical severity

| | |
|---|---|
| Test Case ID | report_service_alerts_with_critical_severity |
| Description | Verify that the critical alerts for services show up in the Grafana and Nagios UI. |
| Prerequisites | Environment deployed with the 4 plugins (see *Deploy an environment with the plugins*). |

#### Steps

1. Open the Grafana URL at `http://<IP address of the 'lma' node>:8000/` and load the Nova dashboard.
2. Open the Nagios URL at `http://<IP address of the 'lma' node>:8001/` in another tab and click the 'Services' menu item.
3. Connect to one of the controller nodes using ssh and stop the nova-api service.
4. Connect to a second controller node using ssh and stop the nova-api service.
5. Wait for at least 1 minute.
6. On Grafana, check the following items:
    (a) the box in the upper left corner of the dashboard displays 'CRIT' with a red background,
    (b) the API panels report 2 entities as down.
7. On Nagios, check the following items:
    (a) the 'nova' service is in 'CRITICAL' state,

   (b) the local user root on the lma node has received an email about the service being in critical state.
8. Restart the nova-api service on both nodes.
9. Wait for at least 1 minute.
10. On Grafana, check the following items:
    (a) the box in the upper left corner of the dashboard displays 'OKAY' with an green background,
    (b) the API panels report 0 entity as down.
11. On Nagios, check the following items:
    (a) the 'nova' service is in 'OK' state,
    (b) the local user root on the lma node has received an email about the recovery of the service.
12. Connect to one of the controller nodes using ssh and stop the nova-scheduler service.
13. Connect to a second controller node using ssh and stop the nova-scheduler service.
14. Wait for at least 3 minutes.
15. On Grafana, check the following items:
    (a) the box in the upper left corner of the dashboard displays 'CRIT' with a red background,
    (b) the scheduler panel reports 2 entities as down.
16. On Nagios, check the following items:
    (a) the 'nova' service is in 'CRITICAL' state,
    (b) the local user root on the lma node has received an email about the service being in critical state.
17. Restart the nova-scheduler service on both nodes.
18. Wait for at least 1 minute.
19. On Grafana, check the following items:
    (a) the box in the upper left corner of the dashboard displays 'OKAY' with an green background,
    (b) the scheduler panel reports 0 entity as down.
20. On Nagios, check the following items:
    (a) the 'nova' service is in 'OK' state,
    (b) the local user root on the lma node has received an email about the recovery of the service.
21. Repeat steps 2 to 21 for the following services:
    (a) Cinder (stopping and starting the cinder-api and cinder-scheduler services respectively).
    (b) Neutron (stopping and starting the neutron-server and neutron-openvswitch-agent services respectively).
22. Repeat steps 2 to 11 for the following services:
    (a) Glance (stopping and starting the glance-api service).
    (b) Heat (stopping and starting the heat-api service).
    (c) Keystone (stopping and starting the Apache service).

#### Expected Result

The Grafana UI shows that the global service status goes from ok to critical and back to ok. It also reports detailed information about which entity is missing.

The Nagios UI shows that the service status goes from ok to critical and back to ok. Alerts are sent by email to the configured recipient.

### 1.3.12 Report node alerts with warning severity

| | |
|---|---|
| Test Case ID | report_node_alerts_with_warning_severity |
| Description | Verify that the warning alerts for nodes show up in the Grafana and Nagios UI. |
| Prerequisites | Environment deployed with the 4 plugins (see *Deploy an environment with the plugins*). |

#### Steps

1. Open the Grafana URL at `http://<IP address of the 'lma' node>:8000/` and load the MySQL dashboard.
2. Open the Nagios URL at `http://<IP address of the 'lma' node>:8001/` in another tab and click the 'Services' menu item.
3. Connect to one of the controller nodes using ssh and run:

```
fallocate -l $(df | grep /dev/mapper/mysql-root | awk '{ printf("%.0f\n", 1024 * (($3 + $4) * 9
```

4. Wait for at least 1 minute.
5. On Grafana, check the following items:
   (a) the box in the upper left corner of the dashboard displays 'OKAY' with an green background,
6. On Nagios, check the following items:
   (a) the 'mysql' service is in 'OK' state,
   (b) the 'mysql-nodes.mysql-fs' service is in 'WARNING' state for the node.
7. Connect to a second controller node using ssh and run:

```
fallocate -l $(df | grep /dev/mapper/mysql-root | awk '{ printf("%.0f\n", 1024 * (($3 + $4) * 9
```

8. Wait for at least 1 minute.
9. On Grafana, check the following items:
   (a) the box in the upper left corner of the dashboard displays 'WARN' with an orange background,
   (b) an annotation telling that the service went from 'OKAY' to 'WARN' is displayed.
10. On Nagios, check the following items:
    (a) the 'mysql' service is in 'WARNING' state,
    (b) the 'mysql-nodes.mysql-fs' service is in 'WARNING' state for the 2 nodes,
    (c) the local user root on the lma node has received an email about the service being in warning state.
11. Run the following command on both controller nodes:

```
rm /var/lib/mysql/test
```

12. Wait for at least 1 minute.
13. On Grafana, check the following items:
    (a) the box in the upper left corner of the dashboard displays 'OKAY' with an green background,
    (b) an annotation telling that the service went from 'WARN' to 'OKAY' is displayed.
14. On Nagios, check the following items:
    (a) the 'mysql' service is in 'OK' state,
    (b) the 'mysql-nodes.mysql-fs' service is in 'OKAY' state for the 2 nodes,
    (c) the local user root on the lma node has received an email about the recovery of the service.

#### Expected Result

The Grafana UI shows that the global 'mysql' status goes from ok to warning and back to ok. It also reports detailed information about the problem in the annotations.

The Nagios UI shows that the service status goes from ok to warning and back to ok. Alerts are sent by email to the configured recipient.

### 1.3.13 Report node alerts with critical severity

| Test Case ID | report_node_alerts_with_critical_severity |
|---|---|
| Description | Verify that the critical alerts for nodes show up in the Grafana and Nagios UI. |
| Prerequisites | Environment deployed with the 4 plugins (see *Deploy an environment with the plugins*). |

#### Steps

1. Open the Grafana URL at `http://<IP address of the 'lma' node>:8000/` and load the MySQL dashboard.
2. Open the Nagios URL at `http://<IP address of the 'lma' node>:8001/` in another tab and click the 'Services' menu item.
3. Connect to one of the controller nodes using ssh and run:

```
fallocate -l $(df | grep /dev/mapper/mysql-root | awk '{ printf("%.0f\n", 1024 * (($3 + $4) * 9
```

4. Wait for at least 1 minute.
5. On Grafana, check the following items:
    (a) the box in the upper left corner of the dashboard displays 'OKAY' with an green background,
6. On Nagios, check the following items:
    (a) the 'mysql' service is in 'OK' state,
    (b) the 'mysql-nodes.mysql-fs' service is in 'CRITICAL' state for the node.
7. Connect to a second controller node using ssh and run:

```
fallocate -l $(df | grep /dev/mapper/mysql-root | awk '{ printf("%.0f\n", 1024 * (($3 + $4) * 9
```

8. Wait for at least 1 minute.
9. On Grafana, check the following items:

   (a) the box in the upper left corner of the dashboard displays 'CRIT' with an red background,

   (b) an annotation telling that the service went from 'OKAY' to 'CRIT' is displayed.

10. On Nagios, check the following items:

    (a) the 'mysql' service is in 'CRITICAL' state,

    (b) the 'mysql-nodes.mysql-fs' service is in 'CRITICAL' state for the 2 nodes,

    (c) the local user root on the lma node has received an email about the service being in critical state.

11. Run the following command on both controller nodes:

```
rm /var/lib/mysql/test
```

12. Wait for at least 1 minute.

13. On Grafana, check the following items:

    (a) the box in the upper left corner of the dashboard displays 'OKAY' with an green background,

    (b) an annotation telling that the service went from 'CRIT' to 'OKAY' is displayed.

14. On Nagios, check the following items:

    (a) the 'mysql' service is in 'OK' state,

    (b) the 'mysql-nodes.mysql-fs' service is in 'OKAY' state for the 2 nodes,

    (c) the local user root on the lma node has received an email about the recovery of the service.

#### Expected Result

The Grafana UI shows that the global 'mysql' status goes from ok to critical and back to ok. It also reports detailed information about the problem in the annotations.

The Nagios UI shows that the service status goes from ok to critical and back to ok. Alerts are sent by email to the configured recipient.

## 1.4 Non-functional testing

### 1.4.1 Simulate network failure on the analytics node

| Test Case ID | network_failure_on_analytics_node |
|---|---|
| Description | Verify that the backends and dashboards recover after a network failure. |
| Prerequisites | Environment deployed with the 4 plugins (see *Deploy an environment with the plugins*). |

#### Steps

1. Copy this script to the analytics node:

```
#!/bin/sh
/sbin/iptables -I INPUT -j DROP
sleep 30
/sbin/iptables -D INPUT -j DROP
```

2. Login to the analytics node using SSH

3. Run the script and wait for it to complete.
4. Check that the Kibana, Grafana and Nagios dashboards are available.
5. Check that data continues to be pushed by the various nodes once the network failure has ended.

#### Expected Result

The collectors recover from the network outage of the analytics node.

## 1.5 Appendix

- The LMA Collector documentation.
- The LMA Infrastructure Alerting documentation.
- The Elasticsearch-Kibana documentation.
- The InfluxDB-Grafana documentation.
- The official Kibana documentation.
- The official Grafana documentation.
- The official Nagios documentation.

CHAPTER 2

# Indices and Tables

- search