

OSF

LMA Toolchain

Test Plan for the LMA Fuel Plugins
version 0.7.0

[OSF](#)

[Revision history](#)

[The LMA Collector Plugin](#)

[Developer's specification](#)

[Limitations](#)

[Test strategy](#)

[Acceptance criteria](#)

[Test environment, infrastructure and tools](#)

[Product compatibility matrix](#)

[System testing](#)

[Install the plugins](#)

[The 3 plugins are present in the Fuel UI.](#)

[Deploy an environment with the plugins](#)

[Modifying env with enabled plugin \(removing/adding controller nodes\)](#)

[Modifying env with enabled plugin \(removing/adding compute node\)](#)

[Uninstall of plugin with deployed environment](#)

[Uninstall of plugin](#)

[Functional testing](#)

[Display and query logs in the Kibana UI](#)

[Display and query nova notifications in the Kibana UI](#)

[Display the dashboards in the Grafana UI](#)

[Display the Nova metrics in the Grafana UI](#)

[Display degraded status in the Grafana UI](#)

[Non-functional testing](#)

[Network failure on analytics node](#)

[Appendix](#)

Revision history

Version	Revision date	Editor	Comment
0.1	27.04.2015	Simon Pasquier (spasquier@mirantis.com)	Initial revision.
0.2	25.06.2015	Vadim Rovachev (vrovachev@mirantis.com)	Added more detailed test suites for query notifications.
0.3	29.06.2015	Vadim Rovachev (vrovachev@mirantis.com)	Updated system tests according to new criteria for plugin certification.
0.4	01.07.2015	Simon Pasquier (spasquier@mirantis.com)	Fix prerequisites.

The LMA Collector Plugin

The LMA (Logging, Monitoring & Alerting) collector is a service running on each OpenStack node that collects metrics, logs and notifications. This data can be sent to [Elasticsearch](#) and/or [InfluxDB](#) backends for diagnostic, troubleshooting and alerting purposes.

The LMA Collector plugin is a service running on each OpenStack nodes which:

- Send logs and notifications to Elasticsearch so operators can more easily troubleshoot issues.
- Send metrics to InfluxDB so operators can monitor and diagnose the usage of resources. This will cover:
 - Operating system metrics (CPU, RAM, ...).
 - Service metrics (MySQL, RabbitMQ, ...).
 - OpenStack metrics (for instance, the number of free/used vCPUs).
 - Metrics extracted from logs and notifications (for instance, the HTTP response times).

In conjunction with the LMA Collector plugin, 2 other plugins have been developed to deploy the backends required by the Collector:

- The [Elasticsearch-Kibana](#) plugin
- The InfluxDB-Grafana plugin

In this test plan, we'll assume that the 3 plugins are tested altogether because the Kibana UI and Grafana UI greatly simplify the validation of the processing done by the LMA collector.

Developer's specification

Document	Link
LMA Collector plugin specification	https://github.com/stackforge/fuel-plugin-lma-collector/blob/master/specs/lma-collector-plugin-spec.rst
LMA Collector README file	https://github.com/stackforge/fuel-plugin-lma-collector/blob/master/README.md
Elasticsearch-Kibana plugin specification	https://github.com/stackforge/fuel-plugin-elasticsearch-kibana/blob/master/specs/elasticsearch-kibana-plugin.rst
Elasticsearch-Kibana README file	https://github.com/stackforge/fuel-plugin-elasticsearch-kibana/blob/master/README.md
InfluxDB-Grafana plugin specification	https://github.com/stackforge/fuel-plugin-influxdb-grafana/blob/master/specs/influxdb-grafana-plugin.rst
InfluxDB-Grafana plugin README	https://github.com/stackforge/fuel-plugin-influxdb-grafana

file	a/blob/master/README.md
------	---

Limitations

The plugin is only compatible with OpenStack environments deployed with Neutron for networking.

The ElasticSearch-Kibana and InfluxDB-Grafana plugins shouldn't be installed on a node that has a role different than the 'base-os' role. Deploying any of the above plugins on an OpenStack node that has controller, compute, or storage role is not supported. Similarly, deploying any of the above plugins along with other plugins is not supported. Only the co-deployment of ElasticSearch-Kibana and InfluxDB-Grafana plugins is supported.

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.

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 the 'base-os' nodes .
3. The Elasticsearch server and the Kibana UI are running on a base-os node.
4. The InfluxDB server and the Grafana UI are running on a base-os node.
5. The Kibana UI can be used to index and search both log messages and notifications.
6. The Grafana dashboards display detailed metrics for the main OpenStack services.
7. The plugins can be uninstalled when no environment uses them.

Test environment, infrastructure and tools

Fuel master node with the 3 plugins installed.

In addition to the nodes reserved for OpenStack, an additional node with 3 separate disks is required to deploy the Elasticsearch-Kibana and InfluxDB-Grafana plugins on the same node. The first disk (sda) is used for the root file-system, the second disk (sdb) is used for ElasticSearch and the third disk (sdc) is used for InfluxDB. Alternatively you can use the same disk for both ElasticSearch and InfluxDB but in this case you have to specify the size that is allocated to each.

Product compatibility matrix

Product	Version/Comment
Mirantis OpenStack	6.1 and higher

LMA Collector plugin	0.7.0 and higher
Elasticsearch-Kibana plugin	0.7.0 and higher
InfluxDB-Grafana plugin	0.7.0 and higher

System testing

Install the plugins

Test Case ID	install_lma_plugins
Description	Verify that the plugins can be installed.
Steps	<ol style="list-style-type: none"> 1. Copy the 3 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 Settings tab.
Expected Result	The 3 plugins are present in the Fuel UI.

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.
Steps	<ol style="list-style-type: none"> 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 checkbox and fill-in the required fields: <ol style="list-style-type: none"> a. Enter 'slave-05_base-os' in the 'Elasticsearch node name' input box. b. Select 'Local node' for InfluxDB and enter 'slave-05_base-os' in the 'InfluxDB node name' input box. 5. Select the Elasticsearch-Kibana plugin checkbox and fill-in the required fields:

	<ol style="list-style-type: none"> a. Change the node's name to 'slave-05_base-os'. 6. Select the InfluxDB-Grafana plugin checkbox and fill-in the required fields: <ol style="list-style-type: none"> a. Change the node's name to 'slave-05_base-os'. b. Enter 'lmapass' as the root and user passwords. 7. Click on the Nodes tab of the Fuel web UI. 8. Assign roles to nodes: <ol style="list-style-type: none"> a. 3 nodes with the 'controller' role b. 1 node with the 'compute' + 'cinder' node c. 1 node with the 'base-os' role. 9. Rename the 'base-os' node to 'slave-05_base-os'. 10. Click 'Deploy changes'. 11. Once the deployment has finished, connect to each node of the environment using ssh and run the following checks: <ol style="list-style-type: none"> a. Check that the LMA collector service is running 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 "slave-05_base-os" node>:9200/) d. Check that the node can connect to the Elasticsearch server (http://<IP address of the "slave-05_base-os" node>:8086/) e. Check that the node can connect to the Kibana UI (http://<IP address of the "slave-05_base-os" node>:80/) f. Check that the node can connect to the Grafana UI (http://<IP address of the "slave-05_base-os" node>:8000/)
Expected Result	The environment is deployed successfully.

Modifying env with enabled plugin (removing/adding controller nodes)

Test Case ID	modify_env_with_plugin_remove_add_controller
Description	Verify that the env with plugins can scale (remove/add controller)
Prerequisites	Environment deployed with the 3 plugins (deploy_lma_plugins).
Steps	<ol style="list-style-type: none"> 1. Remove 1 node with the controller role 2. Re-deploy the cluster 3. Check the plugin services using cli 4. Run OSTF 5. Add 1 new node with the controller role 6. Re-deploy cluster

	<ol style="list-style-type: none"> 7. Check the plugin services using cli 8. Run OSTF
Expected Result	<i>OSTF tests pass successfully, and all the plugin services are running and worked as expected after each modification of the environment.</i>

Modifying env with enabled plugin (removing/adding compute node)

Test Case ID	modify_env_with_plugin_remove_add_compute
Description	Verify that the env with plugins can scale (remove/add compute)
Prerequisites	Environment deployed with the 3 plugins (deploy_Ima_plugins).
Steps	<ol style="list-style-type: none"> 1. Add 1 node with the compute role 2. Re-deploy the cluster 3. Check the plugin services using cli 4. Run OSTF 5. Remove 1 node with the compute role 6. Re-deploy cluster 7. Check the plugin services using cli 8. Run OSTF
Expected Result	<i>OSTF tests pass successfully, and all the plugin services are running and worked as expected after each modification of the environment.</i>

Uninstall of plugin with deployed environment

Test Case ID	uninstall_plugin_with_deployed_env
Description	Verify that the plugins can delete with installed env.
Prerequisites	Environment deployed with the 3 plugins (deploy_Ima_plugins).
Steps	<ol style="list-style-type: none"> 1. Try to delete plugins and ensure that present in cli alert: "400 Client Error: Bad Request (Can't delete plugin which is enabled for some environment.)" 2. Remove the environment.

	<ol style="list-style-type: none"> 3. Remove the plugins. 4. Check that it was successfully removed
Expected Result	<i>Alert is present when we try to delete plugins which are attached to enabled environment. When the environment is removed, plugins are removed successfully too.</i>

Uninstall of plugin

Test Case ID	uninstall_plugin
Description	Verify that the plugins successfully uninstalled.
Prerequisites	3 plugins installed on the Fuel node (install_Ima_plugins).
Steps	<ol style="list-style-type: none"> 1. Remove the plugins. 2. Check that they was successfully removed.
Expected Result	<i>Plugins are removed successfully</i>

Functional testing

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 3 plugins (deploy_Ima_plugins).
Steps	<ol style="list-style-type: none"> 1. Open the Kibana URL at http://<IP address of the analytics 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 nodes deployed in the environment (except 'for base-os' node).

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 3 plugins (deploy_Ima_plugins).
Steps	<ol style="list-style-type: none"> 1. Launch, update, rebuild, resize, power off, power on, snapshot, suspend, shutdown, delete an instance in the OpenStack environment (using the Horizon dashboard for example) and write down the instance id. 2. Open the Kibana URL at http://<IP address of the analytics 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 project obtained exclude compute.instance.create.error, compute.instance.resize.revert.{start end}.

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 3 plugins (deploy_Ima_plugins).
Steps	<ol style="list-style-type: none"> 1. Run OSTF platform test "Ceilometer test to check notifications from Glance" 2. Open the Kibana URL at http://<IP address of the analytics node>/ 3. Open the Notifications dashboard using the 'Load' icon. 4. Enter 'glance' in the Query box.
Expected Result	All event types for glance project obtained.

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 3 plugins (deploy_Ima_plugins).
Steps	<ol style="list-style-type: none"> 1. Create and update 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 analytics 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 project obtained.

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 3 plugins (deploy_Ima_plugins).
Steps	<ol style="list-style-type: none"> 1. Run all OSTF heat platform tests. 2. Open the Kibana URL at http://<IP address of the analytics node>/ 3. Open the Notifications dashboard using the 'Load' icon. 4. Enter 'heat' in the Query box.
Expected Result	All event types for heat project obtained.

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 3 plugins (deploy_Ima_plugins).
Steps	<ol style="list-style-type: none"> 1. Run OSTF functional tests: "Create security group" , "Check network connectivity from instance via floating IP". 2. Open the Kibana URL at http://<IP address of the analytics node>/ 3. Open the Notifications dashboard using the 'Load' icon.

	4. Enter 'neutron' in the Query box.
Expected Result	All event types for neutron project obtained.

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 3 plugins (deploy_Ima_plugins).
Steps	<ol style="list-style-type: none"> 1. Run OSTF platform test: "Ceilometer test to check notifications from Keystone" 2. Open the Kibana URL at http://<IP address of the analytics node>/ 3. Open the Notifications dashboard using the 'Load' icon. 4. Enter 'keystone' in the Query box.
Expected Result	All event types for keystone project obtained.

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 3 plugins (deploy_Ima_plugins).
Steps	<ol style="list-style-type: none"> 1. Open the Grafana URL at http://<IP address of the 'analytics' node>:8000/ and verify that the dashboard is ok (all boxes are green). 2. Click the 'Cinder' box and verify that the dashboard is ok. 3. Go back and repeat step 3 for the following dashboards: <ol style="list-style-type: none"> a. Glance b. Heat c. Keystone d. Nova e. Neutron f. RabbitMQ g. MySQL h. Apache i. Memcached

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

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 3 plugins (deploy_Ima_plugins).
Steps	<ol style="list-style-type: none"> 1. Open the Grafana URL at http://<IP address of the 'analytics' node>:8000/ 2. Load the 'Nova' dashboard. 3. Connect to the Fuel web UI, launch the full suite of OSTF tests and wait for their completion. 4. Check that the 'instance creation time' graph reports values.
Expected Result	The Grafana UI shows the instance creation time over time.

Display degraded status in the Grafana UI

Test Case ID	display_degraded_status_in_grafana_ui
Description	Verify that the Grafana UI reports degraded status when a service isn't fully operational
Prerequisites	Environment deployed with the 3 plugins (deploy_Ima_plugins).
Steps	<ol style="list-style-type: none"> 1. Open the Grafana URL at http://<IP address of the 'analytics' node>:8000/ 2. Load the 'Nova' dashboard. 3. Connect to one of the controller nodes using ssh and stop the nova-api service. 4. Wait for at least 1 minute, check that the box in the upper left corner of the dashboard displays 'Degraded' with an orange background and that the API boxes report 1 entity as down. 5. Start again the nova-api service. 6. Wait for at least 1 minute, check that the box in the upper left corner of the dashboard displays 'OK' with a green background and that the API boxes report 0 entity as down. 7. Stop the nova-scheduler service.

	<ol style="list-style-type: none"> 8. Wait for at least 3 minutes, check that the box in the upper left corner of the dashboard displays 'Degraded' with an orange background and that the scheduler box report 1 entity as down. 9. Start again the nova-scheduler service. 10. Wait for at least 1 minute, check that the box in the upper left corner of the dashboard displays 'OK' with a green background and that the scheduler box report 0 entity as down. 11. Repeat steps 2 to 10 for the following dashboards: <ol style="list-style-type: none"> 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). 12. Repeat steps 2 to 6 for the following dashboards: <ol style="list-style-type: none"> a. Glance (stopping and starting the glance-api service). b. Heat (stopping and starting the heat-api service). c. Keystone (stopping and starting the keystone-all service).
Expected Result	The Grafana UI shows that the global service status goes from degraded to ok and reports detailed information about which entity is missing.

Non-functional testing

Network failure on 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 3 plugins (deploy_Ima_plugins).
Steps	<ol style="list-style-type: none"> 1. Copy this script to the analytics node <pre>#!/bin/sh /sbin/iptables -I INPUT -j DROP sleep 30 /sbin/iptables -D INPUT -j DROP</pre> 2. Login to the analytics node using SSH 3. Run the script and wait for it to complete. 4. Check that the Kibana and Grafana 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.

Appendix

№	Resource title
1	Grafana documentation: http://docs.grafana.org/guides/gettingstarted/#dashboard
2	Kibana documentation http://www.elastic.co/guide/en/kibana/3.0/index.html
3	Event-type List https://docs.google.com/spreadsheets/d/1ES_hWWLpn_eAur2N1FPNyq_QAs5U36fQOcuCxRZjHESY/edit#gid=0