

# **Test Plan for the Calico Fuel Plugin ver 1.0.0**

## Contents

[Calico Plugin](#)

[Developer's specification](#)

[Limitations](#)

[Test strategy](#)

[Acceptance criteria](#)

[Test environment, infrastructure and tools](#)

[Product compatibility matrix](#)

[Type of testing](#)

[Build Plugin](#)

[Install Plugin](#)

[Verify Calico option in Fuel web UI](#)

[Deploy OpenStack with Calico](#)

[Verify BGP Sessions](#)

[Create VMs](#)

[Test connectivity](#)

[Test security](#)

[Test Initial Route Reflector Configuration](#)

[Test Route Reflector Configuration Changes](#)

[Mandatory Tests:](#)

[Install plugin and deploy environment](#)

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

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

[Uninstall of plugin with deployed environment](#)

[Uninstall of plugin](#)

[Appendix](#)

[Revision history](#)

## Calico Plugin

Calico presents a new approach to virtual networking, based on the same scalable IP networking principles as the Internet. It targets data centers where most of the workloads (VMs, containers or bare metal servers) only require IP connectivity, and provides that using standard IP routing. Isolation between workloads - whether according to tenant ownership, or any finer grained policy - is achieved by iptables programming at the servers hosting the source and destination workloads.

## Developer's specification

Is available on [GitHub](#).

### Limitations

This version of the Calico plugin can only be used in deployments with a single controller node. Multiple controllers will be supported in future.

### Test strategy

At present, all plugin-specific tests are manual, and are concerned with establishing basic Calico function. The Calico project has a large number of manual and automated tests which cover its function, security and performance – this test plan does not replicate those tests.

### Acceptance criteria

All tests should pass.

### Test environment, infrastructure and tools

All tests run in a single environment. This is a Mirantis OpenStack cluster with a Fuel master node, one controller node, and two or more compute nodes. The cluster should be deployed with the Calico plugin enabled, as follows:

1. Create a new OpenStack environment, selecting:
  - Juno on Ubuntu Trusty
  - "Neutron with VLAN segmentation" as the networking setup.
2. Under the settings tab, make sure the following options are checked:
  - "Assign public network to all nodes"
  - "Use Calico Virtual Networking"

3. Under the network tab, configure the 'Public' settings (leaving all of the other sections with their default values):
  - IP Range - 172.18.203.60 - 172.18.203.69
  - CIDR - 172.18.203.0/24
  - Use VLAN tagging: No
  - Gateway - 172.18.203.1
  - Floating IP range - 172.18.203.70 - 172.18.203.79
4. Add one controller and two compute nodes.
5. Deploy changes.

Once the cluster has deployed, go to Project->Network->Networks in the OpenStack web UI and create a network and subnet from which instance IP addresses will be allocated. Use the following settings:

- Name: demo
- IP subnet: 10.65.0.0/24
- Gateway: 10.65.0.1
- DHCP-enabled: Yes

Also in the OpenStack web UI, under Project->Compute->Access&Security, create two new security groups named 'sg1' and 'sg2', both with description 'test'. For each security group, select 'Manage Rules' and add two new rules using the following settings:

- First Rule:
  - Rule: ALL ICMP
  - Direction: Ingress
  - Remote: Security Group
  - Security Group: <whichever of sg1/sg2 is followed by '(current)'\>
  - Ether Type: IPv4
- Second Rule:
  - Rule: SSH
  - Remote: CIDR
  - CIDR: 0.0.0.0/0

## Product compatibility matrix

The plugin is compatible with MOS 6.1.

## Type of testing

As above, this test plan is concerned with establishing that Calico networking has been successfully deployed. Security, performance, and detailed functional testing are covered by the main Project Calico test plan.

## Build Plugin

Test Case ID	build_plugin
Description	Verify that the plugin builds successfully.
Steps	<ol style="list-style-type: none"><li>1. git clone https://github.com/stackforge/fuel-plugin-calico.git</li><li>2. fpb --build fuel-plugin-calico</li></ol>
Expected Result	Outputs the message 'Plugin is built' and the calico-1.0-1.0.0-0.noarch.rpm package is created in the fuel-plugin-calico directory.

## Install Plugin

Test Case ID	install_plugin
Description	Verify that the plugin installs successfully.
Steps	<ol style="list-style-type: none"><li>1. Copy the plugin package into /tmp on the Fuel Master Node.</li><li>2. SSH onto the Fuel Master Node.</li><li>3. fpb --install /tmp/calico-1.0-1.0.0-0.noarch.rpm</li><li>4. fuel plugins --list</li></ol>
Expected Result	Output from step 4 should be: <pre>id   name     version   package_version --- ----- ----- ----- 1    calico   1.0.0     2.0.0</pre>

## Verify Calico option in Fuel web UI

Test Case ID	verify_calico_in_fuel_web_ui
Description	Verify that the Calico plugin appears in the OpenStack UI.
Steps	<ol style="list-style-type: none"><li>1. Create a new OpenStack environment in the Fuel web UI.</li><li>2. Navigate to the settings tab.</li></ol>
Expected Result	There should be a tick box labelled 'Use Calico Virtual Networking'.

## Deploy OpenStack with Calico

Test Case ID	deploy_openstack_with_calico
Description	Verify that an OpenStack environment can be successfully deployed with the Calico plugin enabled.
Steps	<ol style="list-style-type: none"><li>1. Create a new OpenStack environment in the Fuel web UI and configure/deploy as per the instructions in the 'Test environment, infrastructure and tools' section of this test plan.</li></ol>
Expected Result	'Success' message is displayed in the Fuel web UI. Followed by: <i>'Deployment of environment 'test' is done. Access the OpenStack dashboard (Horizon) at ...'</i>

## Verify BGP Sessions

Test Case ID	verify_bgp_sessions
Description	Verify that there is a BGP route reflector running on the controller node, and that it has established peer connections to the compute nodes.
Steps	<ol style="list-style-type: none"><li>1. SSH onto the controller node from the Fuel master node.</li><li>2. Enter the command 'birdc', followed by 'show protocols all'.</li><li>3. Check the output details show two established BGP sessions - one to each compute node.</li></ol>
Expected Result	There is a running route reflector on the controller node, with established BGP peer connections to the two compute nodes.

## Create VMs

Test Case ID	create_vms
Description	Verify that Calico does not interfere with the creation of new VMs.
Steps	<ol style="list-style-type: none"><li>1. In the OpenStack Web UI, go to Project-&gt;Instances.</li><li>2. Launch a batch of 6 VMs with the following details:<ul style="list-style-type: none"><li>• Flavor: m1.tiny</li><li>• Boot from image: TestVM</li><li>• Under the Networking tab, drag 'demo' into the 'Selected Networks' box.</li><li>• Under the Access &amp; Security tab, select either 'sg1' or 'sg2' as the security group. such that roughly half of the VMs are in each security group.</li></ul></li><li>3. Under Admin -&gt; Instances, verify that:<ul style="list-style-type: none"><li>• the requested 6 VMs (aka instances) have been launched</li><li>• they are distributed roughly evenly across the two compute hosts</li><li>• they have each been assigned an IP address from the range that you configured above (e.g. 10.65.0/24)</li><li>• they reach Active status within about a minute.</li></ul></li></ol>
Expected Result	The batch of VMs are correctly distributed, and activate in a reasonable time.

## Test connectivity

Test Case ID	test_connectivity
Description	Verify that Calico has configured the network routing to allow communication between the VMs.
Steps	<ol style="list-style-type: none"><li>1. Log on to one of the new VMs, for example by clicking on one of the instances and then on its Console tab.</li><li>2. Use 'ping' to verify connectivity to the IP address of each of the other VMs in the same security group.</li></ol>
Expected Result	Pings can successfully be sent to, and received from, all the VMs in the same security group.

## Test security

Test Case ID	test_security
Description	Verify that Calico correctly enforces the configured security rules.
Steps	<ol style="list-style-type: none"><li>1. Log on to one of the new VMs, for example by clicking on one of the instances and then on its Console tab.</li><li>2. Use 'ping' to verify lack of connectivity to the IP address of each of the other VMs in the other security group.</li></ol>
Expected Result	Pings can not be sent to, or received from, VMs in the other security group.

## Test Initial Route Reflector Configuration

Test Case ID	test_initial_rr_config
Description	Verify that BIRD's BGP peer configuration is correct.
Steps	<ol style="list-style-type: none"><li>1. Deploy an environment with 1 controller, 1 compute node and one storage node.</li><li>2. Verify that the BIRD instance on the controller is configured with only one peer (the compute node).</li></ol>
Expected Result	BGP peer configuration is created only for compute nodes.

## Test Route Reflector Configuration Changes

Test Case ID	test_rr_config_changes
Description	Verify that BIRD's BGP peer configuration is updated correctly after a change to the deployment.
Steps	<ol style="list-style-type: none"><li>1. Deploy an environment with 1 controller, 1 compute node and one storage node.</li><li>2. Verify that the BIRD instance on the controller is configured with only one peer (the compute node).</li><li>3. Delete the storage node and re-deploy.</li><li>4. Add a compute node and re-deploy.</li></ol>



	<ol style="list-style-type: none"> <li>5. Verify that the BIRD instance on the controller is now configured with two peers (both compute nodes).</li> <li>6. Delete both controllers and re-deploy.</li> <li>7. Add a storage node and re-deploy.</li> <li>8. Verify that the BIRD instance on the controller node is now configured with no peers.</li> </ol>
Expected Result	New BGP peer configuration is added to the BIRD instance on the controller when a compute node is added to the deployment. When a compute node is deleted, the BGP peer config for it is removed the next time that a node is added to the deployment.

## Mandatory Tests:

### Install plugin and deploy environment

Covered above.

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

N/A - current calico plugin only supports deployments with a single controller.

### Modifying env with enabled plugin (removing/adding compute nodes)

Test Case ID	modify_env_with_plugin_remove_add_compute
Steps	<ol style="list-style-type: none"> <li>1. Install the Calico plugin on the fuel master node.</li> <li>2. Using the fuel UI, create an environment with the Calico plugin enabled, editing the network and settings configuration as above.</li> <li>3. Add 1 Controller node and 2 compute nodes.</li> <li>4. Deploy the cluster.</li> <li>5. Run the 'Create VMs', 'Test connectivity' and 'Test security' tests above - all should pass.</li> <li>6. Terminate the created VM instances.</li> <li>7. Remove 1 compute node.</li> <li>8. Re-deploy the cluster.</li> <li>9. Run the 'Create VMs', 'Test connectivity' and 'Test security' tests above - all should pass (note all VMs will be created on the same compute node as there is only one).</li> <li>10. Terminate the created VM instances.</li> <li>11. Add 1 compute node.</li> </ol>

	<p>12. Re-deploy the cluster.</p> <p>13. Run the 'Create VMs', 'Test connectivity' and 'Test security' tests above - all should pass.</p>
Expected Result	The Calico Plugin is installed successfully, the cluster is created, and all plugin services are enabled and working as expected after modifying the environment.

### Uninstall of plugin with deployed environment

Test Case ID	uninstall_plugin_with_deployed_env
Steps	<ol style="list-style-type: none"> <li>1. Install the Calico plugin.</li> <li>2. As above, deploy an environment with the Calico plugin enabled.</li> <li>3. Run the 'Create VMs', 'Test connectivity' and 'Test security' tests above - all should pass.</li> <li>4. Try to remove the Calico plugin. This should fail with the error message: "400 Client Error: Bad Request (Can't delete plugin which is enabled for some environment.)". Verify that the Calico plugin is still installed and working.</li> <li>5. Remove the environment.</li> <li>6. Remove the Calico plugin.</li> <li>7. Check the Calico plugin was successfully removed.</li> </ol>
Expected Result	Plugin is installed successfully. An error message is present when we attempt to remove a plugin which is attached to an enabled environment (and the plugin is not removed). When the environment is removed, the plugin can be removed successfully.

### Uninstall of plugin

Test Case ID	uninstall_plugin
Steps	<ol style="list-style-type: none"> <li>1. Install the Calico plugin.</li> <li>2. Check that it was installed successfully.</li> </ol>

	<ol style="list-style-type: none"><li>3. Remove the Calico plugin.</li><li>4. Check that it was successfully removed.</li></ol>
Expected Result	Plugin was installed and then removed successfully.

## Appendix

Project Calico - <http://www.projectcalico.org/>

Calico Documentation - <http://docs.projectcalico.org/en/latest/index.html>

Calico GitHub - <https://github.com/projectcalico/calico>

## Revision history

<b>Version</b>	<b>Revision date</b>	<b>Editor</b>	<b>Comment</b>
0.1	23.01.2015	Irina Povolotskaya (ipovolotskaya@mirantis.com)	Created the template structure.
0.2	29.04.2015	Joe Marshall (joemarshall@projectcalico.org)	First draft.
0.3	08.05.2015	Emma Gordon (emma@projectcalico.org)	Additional test cases.
0.4	02.07.2015	Emma Gordon (emma@projectcalico.org)	Added new mandatory test cases for all fuel plugins.
0.5	03.08.2015	Emma Gordon (emma@projectcalico.org)	Added new test cases.