Guide to the Calico Plugin for Fuel ver 1.0.0

Guide to the Calico Plugin for Fuel ver 1.0.0 Calico Plugin Requirements Limitations Installation Guide User Guide Deploying Mirantis OpenStack with Calico Networking Using your Calico networked Mirantis OpenStack deployment Demonstration Setup Frequently Asked Questions How do I setup instances with internet access? On the controller, BIRD lists routes to my instances listed as unreachable - is that a problem? Why do instances in different networks have connectivity? Appendix Revision history

Calico Plugin

Calico provides seamless, scalable, secure Layer 3 Virtual Networking for your Mirantis OpenStack Deployment.

By replacing OpenStack's native networking model, Calico provides efficient, easy to troubleshoot networking, without the complexity and inefficiency of overlay networking models. Calico does not require any additional nodes or Calico specific management – it just works, and gets out of your way!

More details can be found at <u>http://docs.projectcalico.org/en/latest/</u>

Requirements

The Calico plugin is currently only compatible with Mirantis OpenStack 6.1.

Limitations

In the current release, Calico requires a deployment with a single OpenStack controller. The current release of this plugin only supports an Ubuntu OpenStack setup.

These limitations will be lifted in future releases.

Installation Guide

To install the Calico plugin, follow these steps:

- 1. Prepare a clean Fuel Master node, as described by the Mirantis documentation.
- 2. Download the Calico plugin from the Fuel Plugin Catalog.
- 3. Copy the plugin onto the Fuel Master node: scp calico-fuel-plugin-1.0-1.0.0-0.noarch.rpm root@<Fuel_Master_Node_IP>:/tmp
- Log into the Fuel Master Node: ssh root@<Fuel Master Node IP>
- 5. Install the plugin:

```
cd /tmp
fuel plugins --install calico-fuel-plugin-1.0-1.0.0-0.noarch.rpm
```

6. Check the plugin was installed correctly by running: fuel plugins --list

The expected output is:

User Guide

Deploying Mirantis OpenStack with Calico Networking

Use the Fuel web UI to deploy an OpenStack cluster, observing the following guidelines:

- 1. Create a new OpenStack environment, selecting:
 - "Juno on Ubuntu Trusty (14.04)" as the <u>distribution</u>
 - "Neutron with VLAN segmentation" as the networking setup
 - All other options can be left as their defaults
- 2. Under the <u>Settings tab</u>, make sure the following options are checked and then save your changes:
 - "Assign public network to all nodes"

~	Assign public network to all nodes
	When disabled, public network will be assigned to controllers and zabbix-server only
C+/	
Sto	brage

• "Use Calico Virtual Networking"



- 3. Under the <u>Networks tab</u>, configure the 'Public' settings (these will need to be set to sensible values for your network setup):
 - IP Range
 - CIDR
 - Use VLAN tagging: No
 - Gateway
 - Floating IP range

All of the other network settings should be left with their default values. Ensure you save your changes once you are finished.

Example network configuration:

Nodes	O Settings	Logs	Health Check	Actions			
Network Setting Neutron with VLAN segm]S entation						
Public							
	Start		End				
IP Range	172.18.20	3.60	172.	172.18.203.69			
CIDR	172.18.20	3.0/24					
Use VLAN tagging							
Gateway	172.18.20	3.1		End			
	Start		End				
Floating IP ranges	172.18.20	3.70	172.	18.203.79			

- 4. Under the Nodes tab, <u>add some nodes</u> (for meaningful testing, you will need at least two compute nodes in addition to the controller). Note that, in this release of Calico, only a single controller node is supported (see <u>Limitations</u> section above).
- 5. <u>Deploy changes</u>.

Using your Calico networked Mirantis OpenStack deployment

You are now ready to use the OpenStack dashboard to configure your deployment. Your particular requirements will determine how you use your OpenStack deployment, but you may wish to refer to the Calico documentation for some common <u>next steps</u>.

Demonstration Setup

The following is a demonstration OpenStack setup that, if wished, can be followed to verify the Calico elements of your OpenStack deployment are operating as intended.

In this example, we will launch a number of VMs (load balanced across the compute hosts), split into two security groups - with VMs in the same security group able to contact each other, but not VMs in the other security group (regardless of which compute host the VMs are on).

Steps:

1. Follow the link from the Fuel web UI to the OpenStack dashboard:



2. Under Project->Network->Networks in the OpenStack dashboard, create a network and subnet from which instance IP addresses will be allocated.

	🗖 openstack		🔳 ad	nin 🗸					
P	Project		N Create Network						
	Compute	×	N				- Cre		
E	Network			Network Subnet * Subnet Detail					
	Network Topology		C	Network Name	Create a new network. In addition a subnet associated with the network can be created in the next panel.				
	Networks		0	Admin State * 😧					
	Routers		C	UP					
	Object Store	×.	Di	- Pc	Next				
	Orchestration	F		« Da	Next :				
A	dmin	Þ							

Use the following settings:

- Network:
 - o Name: demo
 - o Admin State: UP
- Subnet:
 - o Create Subnet: Yes
 - o Name: demo_subnet
 - o Network Address: 10.65.0.0/24
 - o IP Version: IPv4
 - o Gateway IP: 10.65.0.1
- Subnet Detail:
 - o Enable DHCP: Yes
- 3. Under Project->Compute->Access&Security in the OpenStack dashboard, create two new security groups, named 'sg1' and 'sg2', and both with description 'test'.

Openstack	admin +
Project -	A Create Security Group
Compute -	
Overview	Name* Description:
Instances	Security groups are sets of IP filter rules that are applied to the network settings for the VM. After the security group is created, you can add rules to the security group.
Volumes	
Images	Cancel Create Security Group
Access & Security	sg2 test
Network +	Displaying 3 tems

4. For each security group, select 'Manage Rules' and add two new rules.

U openstaci	(La		×	adir
Project		M Add Rule		
Compute	*	S		+ Add Rule
Overview		Rule *	_ Description:	Actions
Instances		Direction	Rules define which traffic is allowed to instances assigned	Delete 5
		Ingress	three main parts:	- Delete (
Volumes		Open Port *	Rule: You can specify the desired rule template or use custom rules, the options are Custom TCP Rule, Custom	Delete F
Images		Port	Open Port/Port Range: For TCP and UDP rules you may	Delete F
Access & Security		Port O	choose to open either a single port or a range of ports. Selecting the "Port Range" option will provide you with	Delete F
Network	×	Di	space to provide both the starting and ending ports for the range. For ICMP rules you instead specify an ICMP type	
Object Store	je.	Remote * 😡	and code in the spaces provided. Remote: You must specify the source of the traffic to be	
Orchestration	1	CIDR	 allowed via this rule. You may do so either in the form of an ID address black (CDD) or via a service group (Security) 	
Orchestration		CIDR 🕑	Group). Selecting a security group as the source will allow	
dmîn	×.	0.0.0/0	any other instance in that security group access to any other instance via this rule.	
dentity	Ъ.			
			Cancel	
			California	

Use 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
- 5. Under Project->Compute->Instances in the OpenStack dashboard, launch several instances.

penstack	🖬 admin 🖌						
Project 👻	Launch Instance						
Compute 👻	h						
Overview	Details * Access & Security * Networking * Post-Creation Advanced Options Availability Zone Specify the details for launching an instance.	c					
Instances	nova The chart below shows the resources used by this						
Volumes	Instance Name * Flavor Details						
Images	Name m1.micro						
Access 9 County	Flavor * O VCPUs 1						
Access & Security	Root Disk 0 GB						
Network 🔸	Instance Count * Ephemeral Disk 0 GB						
Object Store	1 Total Disk 0 GB						
Orchestration >	Instance Boot Source * • RAM 64 MB						
Admin >	Select source Project Limits Number of Instances 0 of No Limit Us	sed					
identity P	Number of VCPUs 0 of No Limit U	sed					
	Total RAM 0 of No Limit MB U	sed					
	Cancel	Launch					

Use the following settings:

- Flavor: m1.tiny
- Instance Boot Source: Boot from Image
- Image Name: TestVM
- Under the Access & Security tab, select one of sg1/sg2 (split your instances roughly 50:50 between the two security groups).
- Under the Networking tab, drag 'demo' into the 'Selected Networks' box.

- 6. Under Admin->Instances in the OpenStack dashboard, verify that:
 - the requested instances have been launched
 - they are distributed roughly evenly across the two compute hosts
 - they have each been assigned an IP address from the range that you configured above (e.g. 10.65.0.0/24)
 - they reach Active status within about a minute.

Project		All	Insta	nces									
Admin 👻		Instances					<u> </u>	Project •		Filter			Filter O Terminate
System	Ŧ		Project	Host	Name	lmage Name	IP Address	Size	Status	Task	Power State	Time since created	Actions
Overview Hypervisors			admin	node-2.datcon.co.uk	TEST3	TestVM	10.65.0.12	m1.tiny	Active	None	Running	23 hours, 54 minutes	Edit Instance 👻
Host Aggregates		۵	admin	node-3.datcon.co.uk	TEST2	TestVM	10.65.0. <mark>1</mark> 1	m1.tiny	Active	None	Running	23 hours, 55 minutes	Edit Instance 👻
Instances			admin	node-2.datcon.co.uk	TEST	TestVM	10.65.0.10	m1.tiny	Active	None	Running	23 hours, 55 minutes	Edit Instance 👻
Volumes			admin	node-2.datcon.co.uk	test3	TestVM	10.65.0.9	m1.tiny	Active	None	Running	23 hours, 56 minutes	Edit Instance 👻
Flavors			admin	node-3.datcon.co.uk	test2	TestVM	10.65.0.8	m1.tiny	Active	None	Running	23 hours, 57 minutes	Edit Instance 👻
Images			admin	node-3.datcon.co.uk	test	TestVM	10.65.0.7	m1.tiny	Active	None	Running	23 hours, 57 minutes	Edit Instance 👻

7. Open a console on one of the instances. You should find that you can ping the other instances in the same security group, but not the instances in the other security group.

D openstack	🗐 admin 👻
Project >	Instance Details: test
Admin	Overview Log Console Action Log
System 👻	
Overview	Instance Console
Hypervisors	If console is not responding to keyboard input: click the grey status bar below. <u>Click here to show only console</u> To exit the fullscreen mode, click the browser's back button.
Host Aggregates	Connected (unencrynfed) fo: OFMU (instance.00000005)
Instances	\$ ping 10.65.0.8 PING 10.65.0.8 (10.65.0.8): 56 data bytes
Volumes	64 bytes from 10.65.0.8; seq=0 tt1=63 time=3.070 ms 64 bytes from 10.65.0.8; seq=1 tt1=63 time=1.014 ms 64 bytes from 10.65.0.8; seq=2 tt1=63 time=0.771 ms
Flavors	10.65.0.8 ping statistics 3 packets transmitted, 3 packets received, 0% packet loss
Images	\$ ping 10.65.0.12 PING 10.65.0.12 (10.65.0.12): 56 data bytes
Networks	10.65.0.12 ping statistics 3 packets transmitted, 0 packets received, 100% packet loss
Routers	\$ _

Frequently Asked Questions

How do I setup instances with internet access?

For outbound access, you need to ensure that your VMs can send traffic to your border gateway router (typically this will be the case, because usually your compute hosts will be able to do so). The border gateway can then perform SNAT.

For inbound connections, you need assign a publically routable IP address to your VM – that is, attach it to a network with a public IP address. You will also need to make sure that your border router (and any intermediate routers between the border router and the compute host) can route to that address too. The simplest way to do that is to peer the border router with the route reflector on the control host.

The Calico documentation has an overview of addressing and connectivity.

On the controller, BIRD lists routes to my instances listed as unreachable - is that a problem?

No, this is expected. On the control node, BIRD is acting as a route reflector, so won't write routes into the Linux forwarding table. Hence these routes are unreachable from the control node. That's ok though – they are reachable from the compute hosts, and therefore from the instances themselves.

Why do instances in different networks have connectivity?

With Calico networking, any two networks will have connectivity, unless you have specifically disabled it using security groups. This is different to standard OpenStack networking – you can find more information in the <u>Calico Neutron API documentation</u>.

Appendix

General Calico docs can be found at <u>http://docs.projectcalico.org/en/latest/</u> The official Calico website is at <u>http://www.projectcalico.org/</u> The Calico code base lives at <u>https://github.com/projectcalico/calico</u>

Revision history

Version	Revision date	Editor	Comment
0.1	04.30.2015	Brook Roberts (brook@projectcalico.org)	Created the document.
0.2	05.07.2015	Emma Gordon (emma@projectcalico.org)	Review markups from Mirantis feedback.
0.3	03.08.2015	Emma Gordon (emma@projectcalico.org)	Updated link to calico repository on GitHub.