The FWaaS Plugin for Fuel Documentation

Release 1.1-1.1.0-1

Mirantis Inc.

January 14, 2016

CONTENTS

CHAPTER

ONE

DOCUMENT PURPOSE

This document provides instructions for installing, configuring and using Neutron Firewall-as-a-Service plugin for Fuel.

1.1 Key terms, acronyms and abbreviations

Term/abbreviation	Definition
FWaaS	Firewall-as-a-Service
IPTables	A user-space application program that allows a system administrator to configure the tables provided by the Linux kernel firewall and the chains and rules it stores. Different kernel modules and programs are currently used for different protocols; IPTables applies to IPv4, ip6tables to IPv6, arptables to ARP, and ebtables to Ethernet frames.
VM	Virtual Machine (Instance)

1.2 FWaaS Plugin

The Firewall-as-a-Service (FWaaS) is a Neutron plugin, which adds perimeter firewall management to Networking. FWaaS uses IPTables to apply firewall policy to the selected router. Whereas security groups operate at the instance-level, FWaaS operates at the router-level.

1.3 Requirements

Requirement	Version/Comment
Fuel	7.0 release with Maintenance Update 2
OpenStack compatibility	2015.1 Kilo with Maintenance Update 2
Operating systems	Ubuntu 14.04 LTS

1.4 Limitations

FWaaS plugin can be enabled only in environments with Neutron with ML2 plugin with OpenVSwitch Mechanism driver (default configuration) as the networking option and tested only with the IPTables driver.

1.5 Known issues

Please make sure that your environment contains maintenance update MU-2 for MOS 7.0 which has a fix for the High bug: [FWaaS] Error firewall state after updating policy or rule [1]

If your environment doesn't contain MU-2, please apply it: How to apply Mirantis OpenStack 7.0 Maintenance Update [2]

[1] https://bugs.launchpad.net/mos/7.0.x/+bug/1510576
[2] https://docs.mirantis.com/openstack/fuel/fuel-7.0/maintenance-updates.html

CHAPTER

TWO

INSTALLATION GUIDE

2.1 Installing FWaaS plugin

1. Download the plugin from Fuel Plugins Catalog [1].
2. Copy the plugin on already installed Fuel Master node:

```
[user@home ~]$ scp fwaas-plugin-1.1-1.1.0-1.noarch.rpm root@:/
<the_Fuel_Master_node_IP>:~/
```

3. Log into the Fuel Master node. Install the plugin:

```
[root@fuel ~]# fuel plugins --install fwaas-plugin-1.1-1.1.0-1.noarch.rpm
```

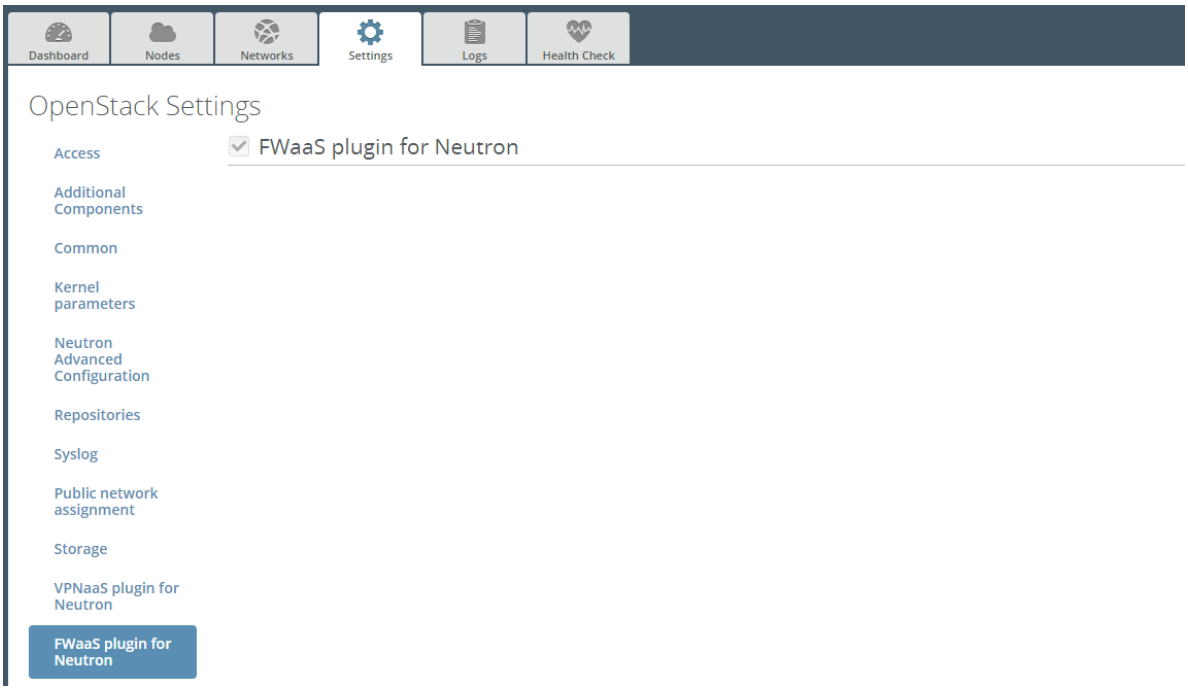
4. Verify that the plugin is installed correctly:

```
[root@fuel ~]# fuel plugins --list
id | name          | version | package_version
---|---------------|---------|----------------
1  | fwaas_plugin | 1.1.0   | 2.0.0
```

2.2 Creating Environment with FWaaS

1. After plugin is installed, create a new OpenStack environment with Neutron as a network provider.
2. Configure your environment [2].
3. Open the Settings tab of the Fuel web UI and scroll down the page. Select FWaaS plugin checkbox:

[1] https://software.mirantis.com/download-mirantis-openstack-fuel-plug-ins
[2] http://docs.mirantis.com/openstack/fuel/fuel-7.0/user-guide.html#configure-your-environment



4. Deploy your environment [3].

2.2.1 References

[3] http://docs.mirantis.com/openstack/fuel/fuel-7.0/user-guide.html#deploy-changes

CHAPTER

THREE

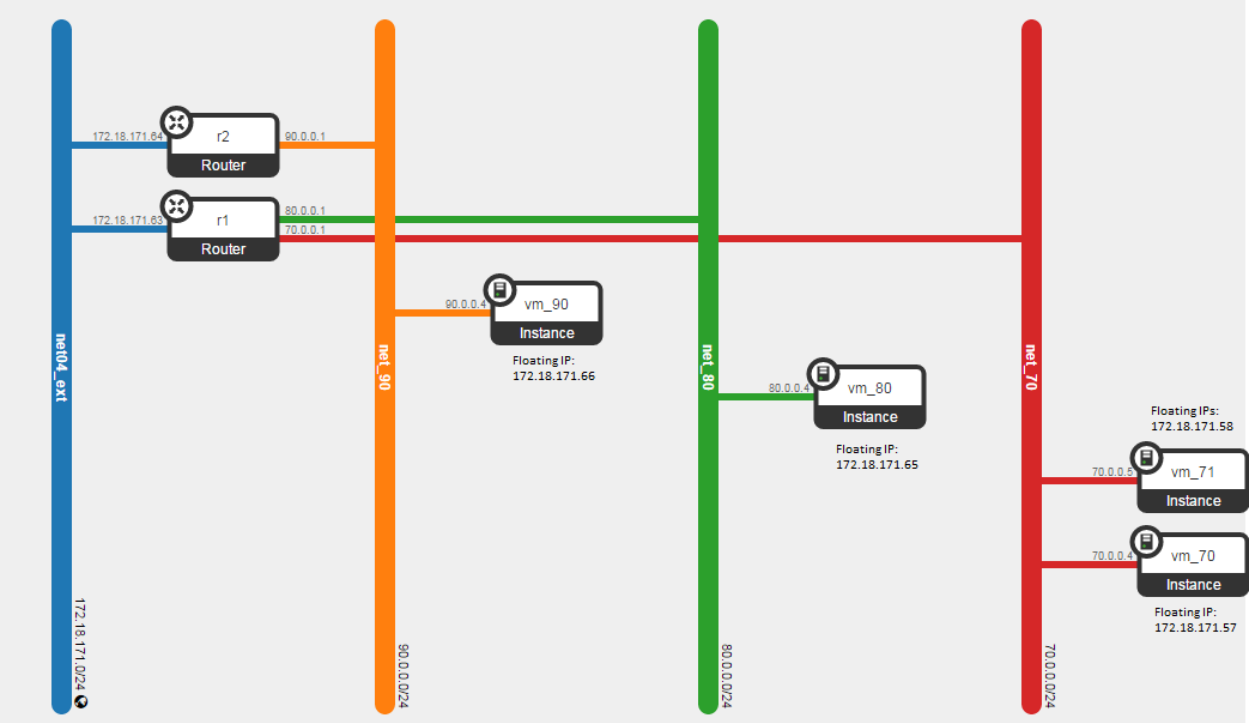
USER GUIDE

3.1 Configuring FWaaS service

Once OpenStack has been deployed, we can start configuring FWaaS.

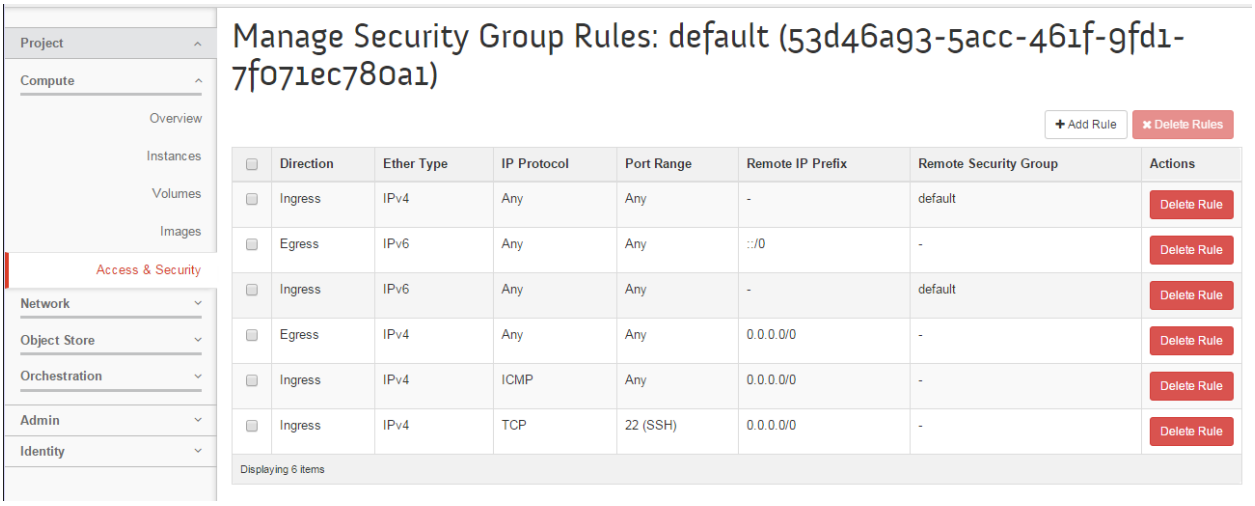
This section provides an example of configuration and step-by-step instructions for configuring the plugin.

Here is an example task. We will have the following network architecture in our Project:



Before we start, we need to be remember that every Project in OpenStack is assigned the default security group for the cluster in its default form, which is usually restrictive. So you'll probably need to create a few additional rules in each Project's default security group: like a general ICMP rule, enabling pings, and a port 22 TCP rule, enabling SSH an example task:

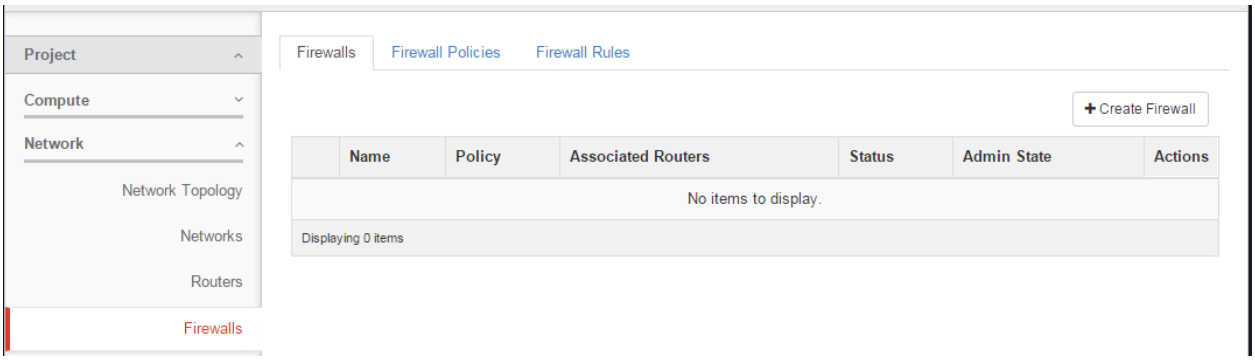
Let's get started with the testing of connectivity between our VMs (using ping). So, for the current state situation is the following (see the network topology above):



	vm_70		vm_71		vm_80		vm_90	
	Local IP 70.0.0.4	Floating IP 172.18.171.57	Local IP 70.0.0.5	Floating IP 172.18.171.58	Local IP 80.0.0.4	Floating IP 172.18.171.65	Local IP 90.0.0.4	Floating IP 172.18.171.66
vm_70	+	+	+	+	+	+	-	+
vm_71	+	+	+	+	+	+	-	+
vm_80	+	+	+	+	+	+	-	+
vm_90	-	+	-	+	-	+	+	+
My PC	-	+	-	+	-	+	-	+

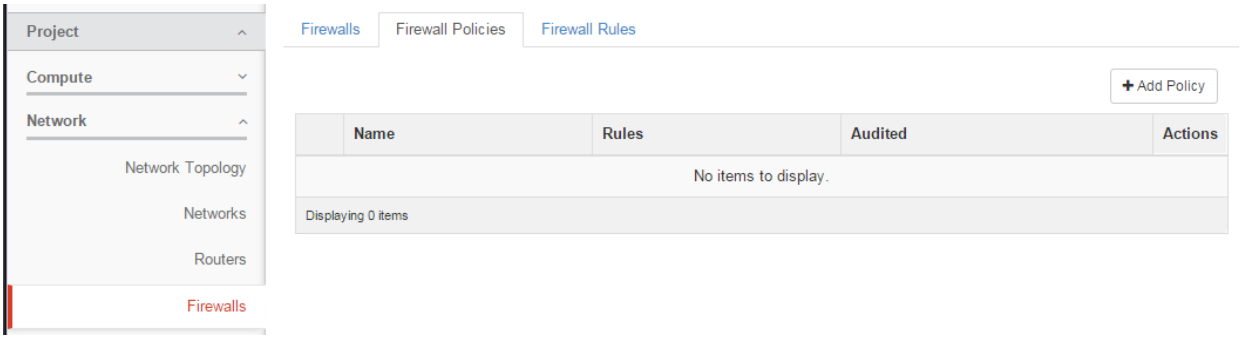
1. Let's create **Firewall**

   Open *Network* menu in the left-hand menu and select *Firewalls* option.



2. Create **Policy**

   Enter *Firewall Policies* tab and click *Add Policy* button.



   In this window, we should fill in policy name and description of this policy in the *Name* and *Description* fields. Also, here we can set *Shared* and *Audited flags*:

   - *Shared* - allow to share your policy with all other Projects.
   - *Audited* - indicate whether the particular firewall policy was audited or not by the creator of the firewall policy.

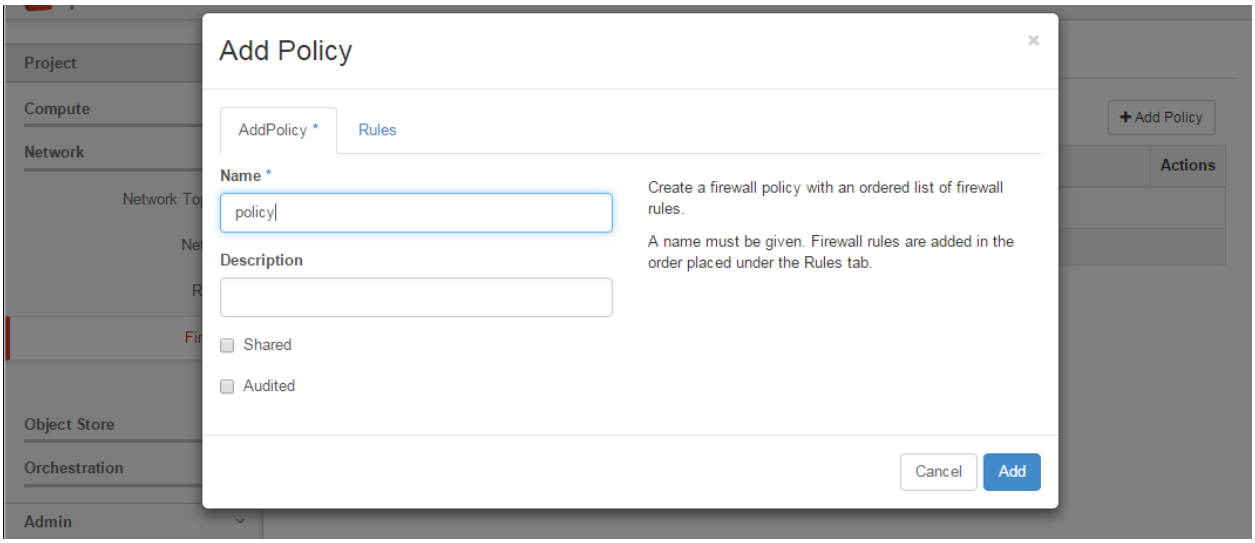
   And click *Add* button to finish.

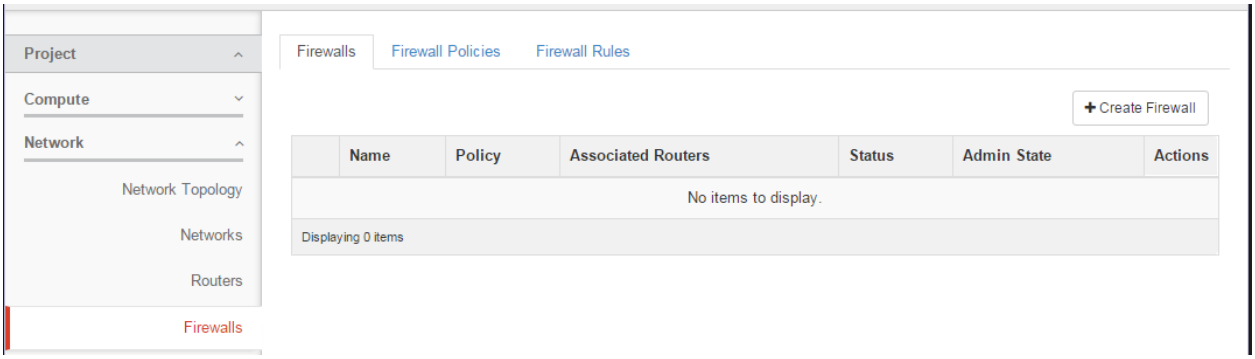
3. Create **Firewall**

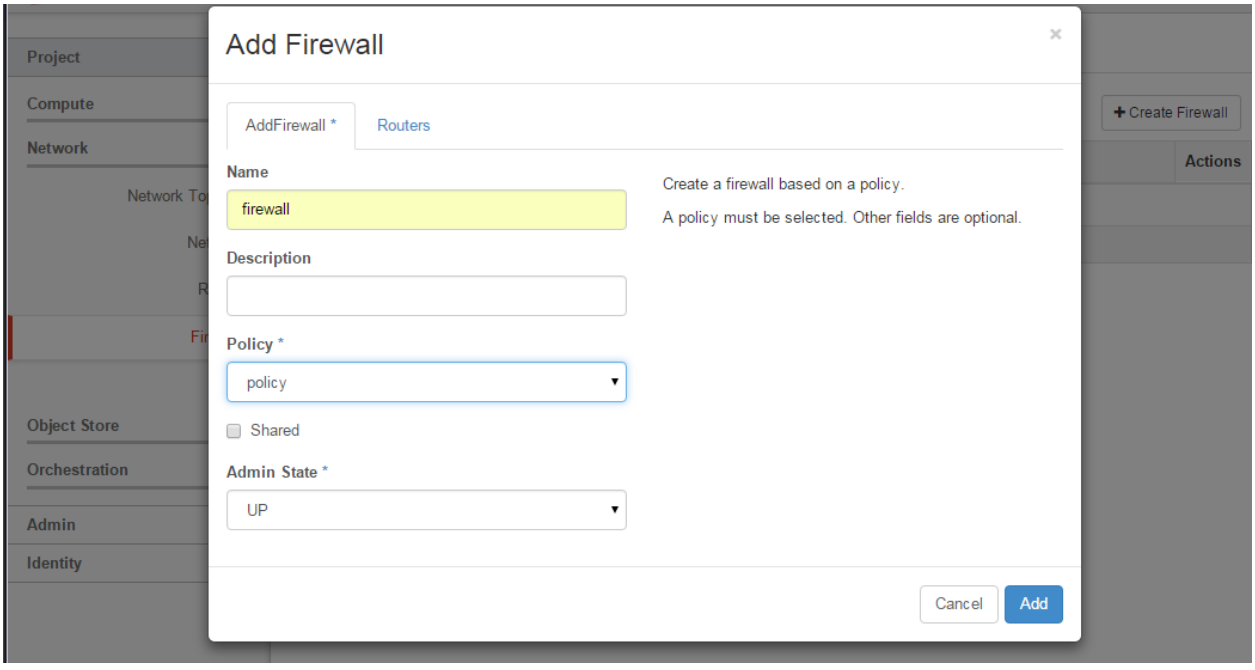
   Enter *Firewalls* tab and click *Create Firewall* button.

   In *Add Firewall* tab we should fill in *Name*, *Description* fields and choose our policy that was created in previous step.

   - *Shared* - allow to share your Firewall with all other Projects.
   - *Admin State* - option provide an ability to set UP or DOWN the Firewall.

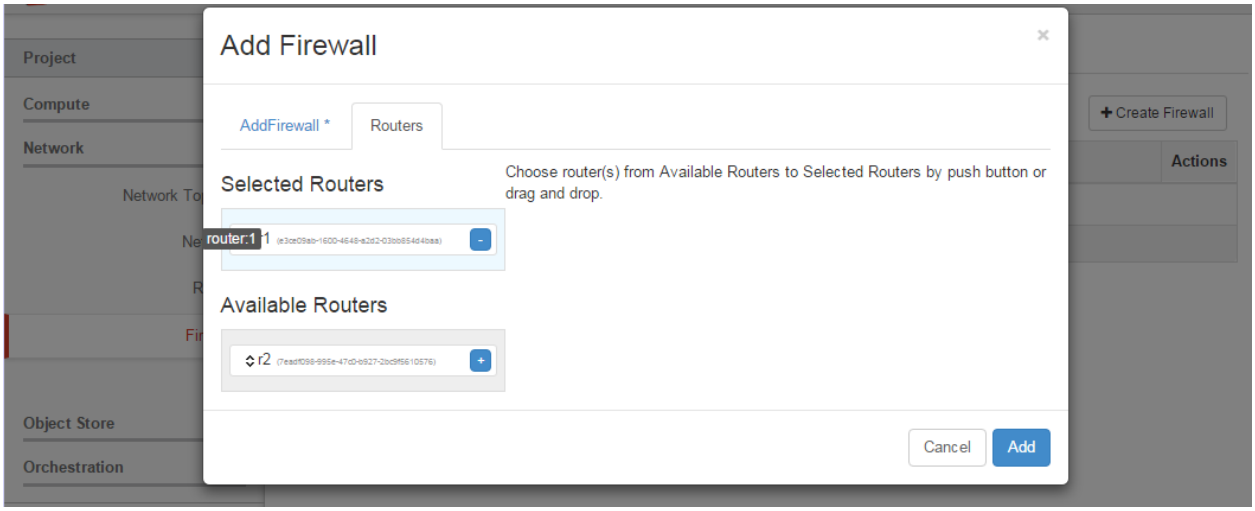

Add Policy
AddPolicy *
Rules
Name *
policy
Description
Shared
Audited
Create a firewall policy with an ordered list of firewall rules.
A name must be given. Firewall rules are added in the order placed under the Rules tab.
Cancel
Add
Project
Compute
Network
Object Store
Orchestration
Admin
+ Add Policy
Actions


Project
Compute
Network
Network Topology
Networks
Routers
Firewalls
Firewalls
Firewall Policies
Firewall Rules
+ Create Firewall
Name
Policy
Associated Routers
Status
Admin State
Actions
No items to display.
Displaying 0 items


Add Firewall
AddFirewall *
Routers
Name
firewall
Description
Policy *
policy
Shared
Admin State *
UP
Create a firewall based on a policy.
A policy must be selected. Other fields are optional.
Cancel
Add
Project
Compute
Network
Object Store
Orchestration
Admin
Identity
+ Create Firewall
Actions

NOTE: The firewall remains in *PENDING_CREATE* state until you create a Networking router and attach an interface to it.

In *Routers* tab we should choose routers from the available routers on which we want to enable our Firewall. Let's apply it only for router **r1**.



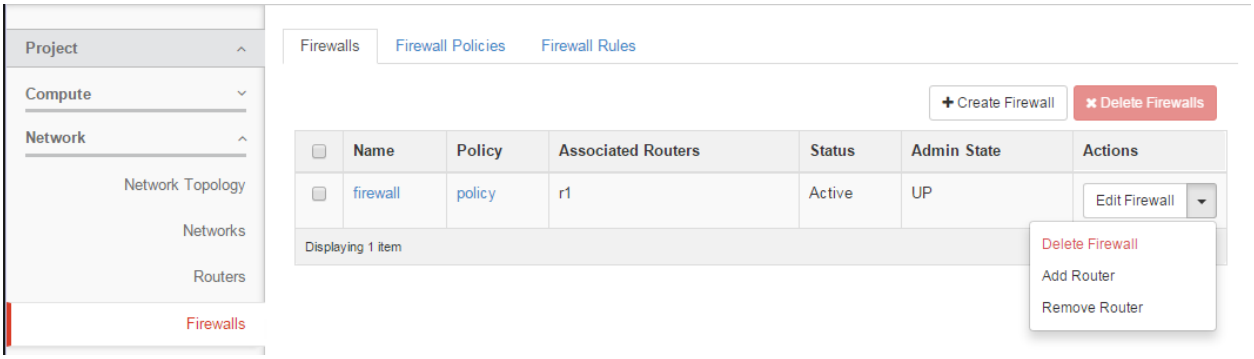
4. Let's test connectivity between our VMs with new Firewall which we applied on the router **r1**

	vm_70		vm_71		vm_80		vm_90	
	Local IP 70.0.0.4	Floating IP 172.18.171.57	Local IP 70.0.0.5	Floating IP 172.18.171.58	Local IP 80.0.0.4	Floating IP 172.18.171.65	Local IP 90.0.0.4	Floating IP 172.18.171.66
vm_70	+	-	+	-	-	-	-	-
vm_71	+	-	+	-	-	-	-	-
vm_80	-	-	-	-	+	-	-	-
vm_90	-	-	-	-	-	-	+	+
My PC	-	-	-	-	-	-	-	+

WARNING: Firewall always adds a default rule to **deny** all at the lowest precedence of each policy. Consequently, a firewall policy with no rules blocks all traffic by default.

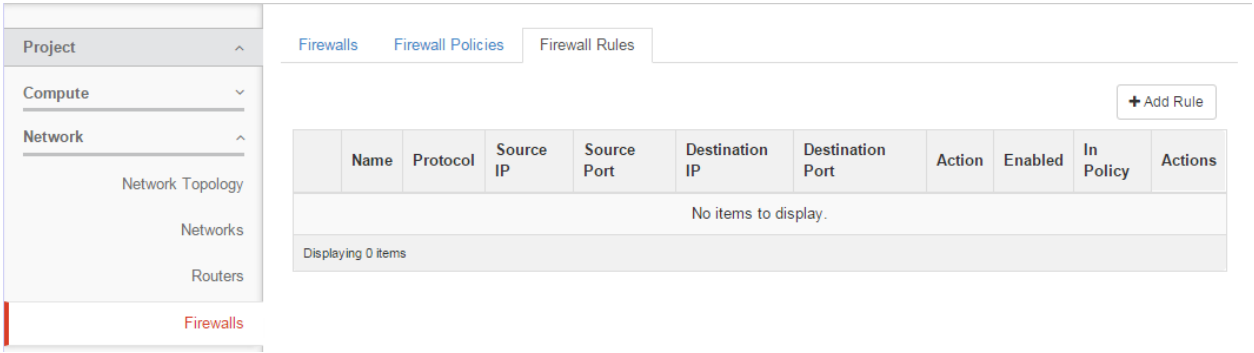
Since we applied our Firewall only for the router **r1** we can that **r1** blocks all traffic and router **r2** works as before. For the adding and removing routers to the Firewall we should click drop-down

button near the *Edit Firewall* button and select *Add/Remove Router*:



5. Create **Rule**

   For the allowing ICMP traffic we need to create a new rule. Enter *Firewall Rules* tab and press *Add Rule* button:



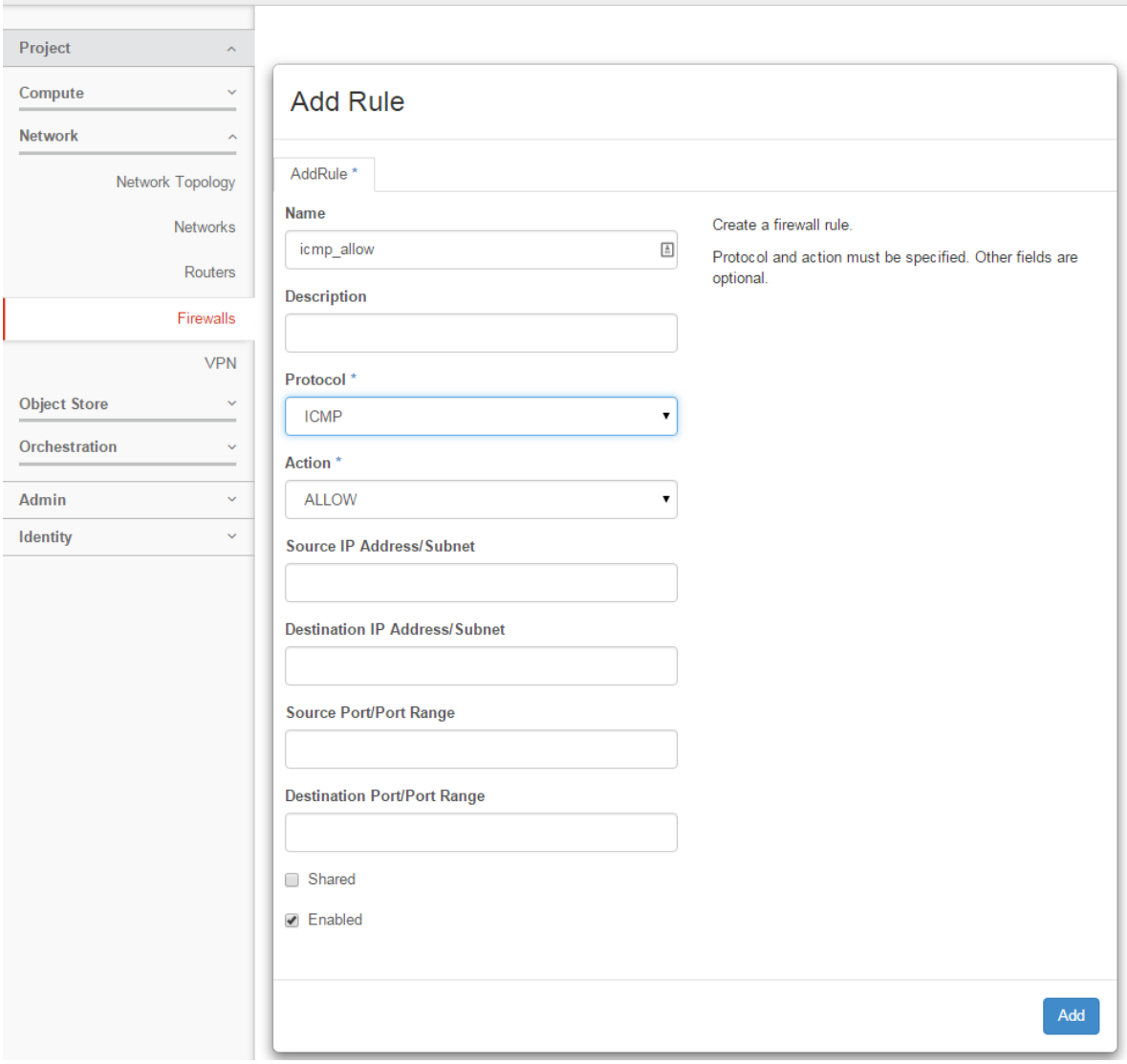
   Here, as usual we should fill in Name and Description fields. And specify the type of traffic, a couple of flags and action for it:

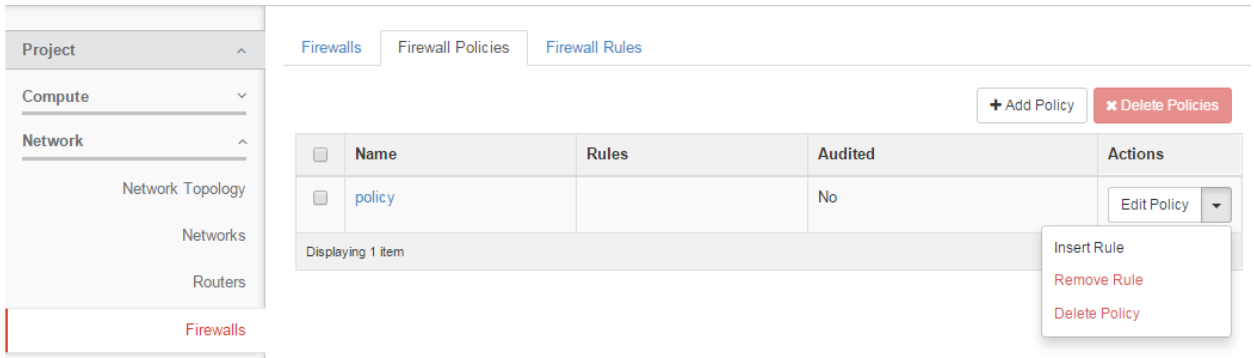
   - *Protocol* - type of protocol (ICMP, TCP, UDP or ANY).
   - *Source(Destination) IP Address/Subnet* - It might be single IP 172.18.161.10 or CIDR like 172.18.161.0/24
   - *Source(Destination) Port / Port Range* - It might be a single Port 80 or range like 100:200.
   - *Action* - what to do (ALLOW or DENY) with this type traffic.
   - *Shared* - allow to share your rule with all other Projects.
   - *Enable* - provide an ability to turn ON or OFF this rule.

6. Add **Rule** to the **Policy**

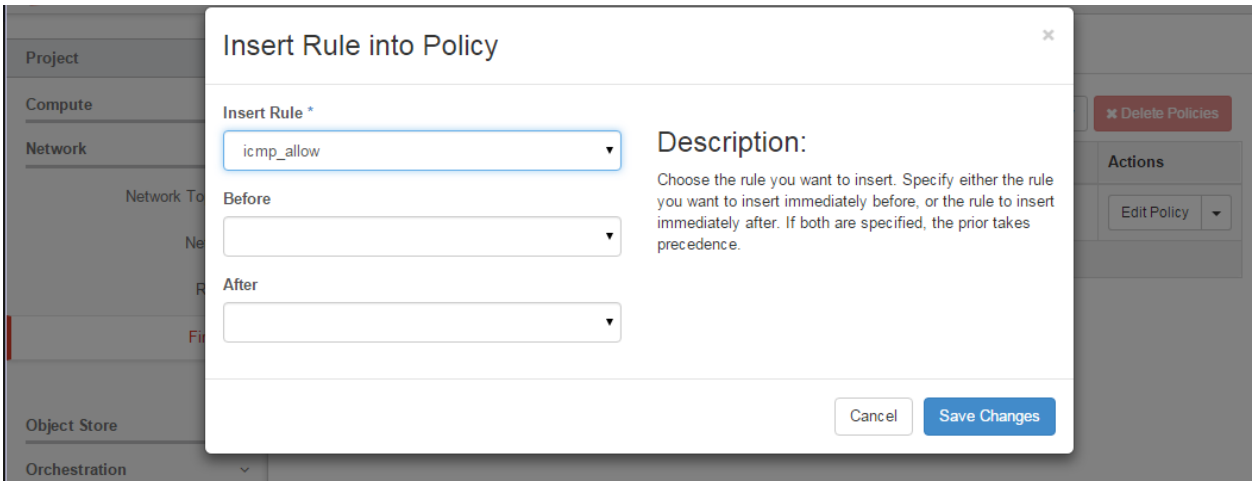
   Add the created rule into our policy:

   - Enter Firewall Policies.
   - In column for our policy, click drop-down button and select Insert Rule.


Project
Compute
Network
Network Topology
Networks
Routers
Firewalls
VPN
Object Store
Orchestration
Admin
Identity
Add Rule
AddRule *
Name
icmp_allow
Description
Protocol *
ICMP
Action *
ALLOW
Source IP Address/Subnet
Destination IP Address/Subnet
Source Port/Port Range
Destination Port/Port Range
Shared
Enabled
Create a firewall rule.
Protocol and action must be specified. Other fields are optional.
Add


Project
Compute
Network
Network Topology
Networks
Routers
Firewalls
Firewalls
Firewall Policies
Firewall Rules
+ Add Policy
✖ Delete Policies
Name
Rules
Audited
Actions
policy
No
Edit Policy
Displaying 1 item
Insert Rule
Remove Rule
Delete Policy

- In *Insert Rule to Policy* window, we can choose the necessary rule and specify the order of applying the rules. It's important that the rules are setup in proper order. The first rule that matches the type of traffic will be used.



7. And let's test connectivity again

	vm_70		vm_71		vm_80		vm_90	
	Local IP 70.0.0.4	Floating IP 172.18.171.57	Local IP 70.0.0.5	Floating IP 172.18.171.58	Local IP 80.0.0.4	Floating IP 172.18.171.65	Local IP 90.0.0.4	Floating IP 172.18.171.66
vm_70	+	+	+	+	+	+	-	+
vm_71	+	+	+	+	+	+	-	+
vm_80	+	+	+	+	+	+	-	+
vm_90	-	+	-	+	-	+	+	+
My PC	-	+	-	+	-	+	-	+

The situation is the same that we have without a Firewall, but only for the ICMP traffic while for the other types of packets it remained the same as at the beginning.

CHAPTER

FOUR

APPENDIX

#	Title of resource	Link on resource
1	Fuel Plugins CLI	Link
2	Firewall-as-a-Service	Link

CHAPTER

FIVE

INDICES AND TABLES

- search