Guide to the FWaaS Plugin version 1.0.0 for Fuel

Revision history

Version	Revision date	Editor	Comment
0.1	02.19.2015	Irina Povolotskaya (ipovolotskaya@mirantis.com)	Created the document structure.
0.2	03.02.2015	Irina Povolotskaya (ipovolotskaya@mirantis.com)	Edited Configuring FWaaS plugin section.
0.3	03.30.2015	Irina Povolotskaya (ipovolotskaya@mirantis.com)	Added Document purpose and Key terms, acronyms and abbreviations sections.
1.0	04.08.2015	Andrey Epifanov (aepifanov@miratnis.com)	Major version

Document purpose

This document provides instructions for installing, configuring and using FWaaS plugin for Fuel.

Key terms, acronyms and abbreviations

Term/acronym/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

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 all Networking routers within a project. FWaaS supports one firewall policy and logical firewall instance per project.
Whereas security groups operate at the instance-level, FWaaS operates at the perimeter to filter traffic at the neutron router.

Requirements

Requirement	Version/Comment
Fuel	6.x release series
OpenStack compatibility	2014.2 Juno
Operating systems	Ubuntu 14.04 LTS CentOS 6.5

Limitations

FWaaS plugin can be enabled only in environments with Neutron as the networking option.

Installation Guide

Installing FWaaS plugin

1. Download the plugin from Fuel Plugins Catalog.

2. Copy the plugin on already installed Fuel Master node. If you do not have the Fuel Master node yet, see Quick Start Guide:

```
scp fwaas-plugin-1.0-1.0.0-0.noarch.rpm root@:<the_Fuel_Master_node_IP>:/tmp
```

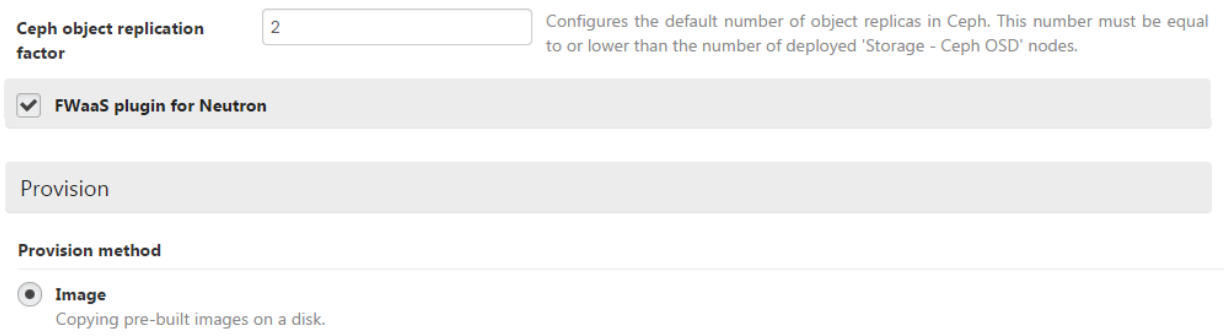
3. Log into the Fuel Master node. Install the plugin:

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

4. After plugin is installed, create a new OpenStack environment with Neutron.

5. Configure your environment.

6. Open the *Settings* tab of the Fuel web UI and scroll down the page. Select FWaaS plugin checkbox:



7. Deploy your environment.

User Guide

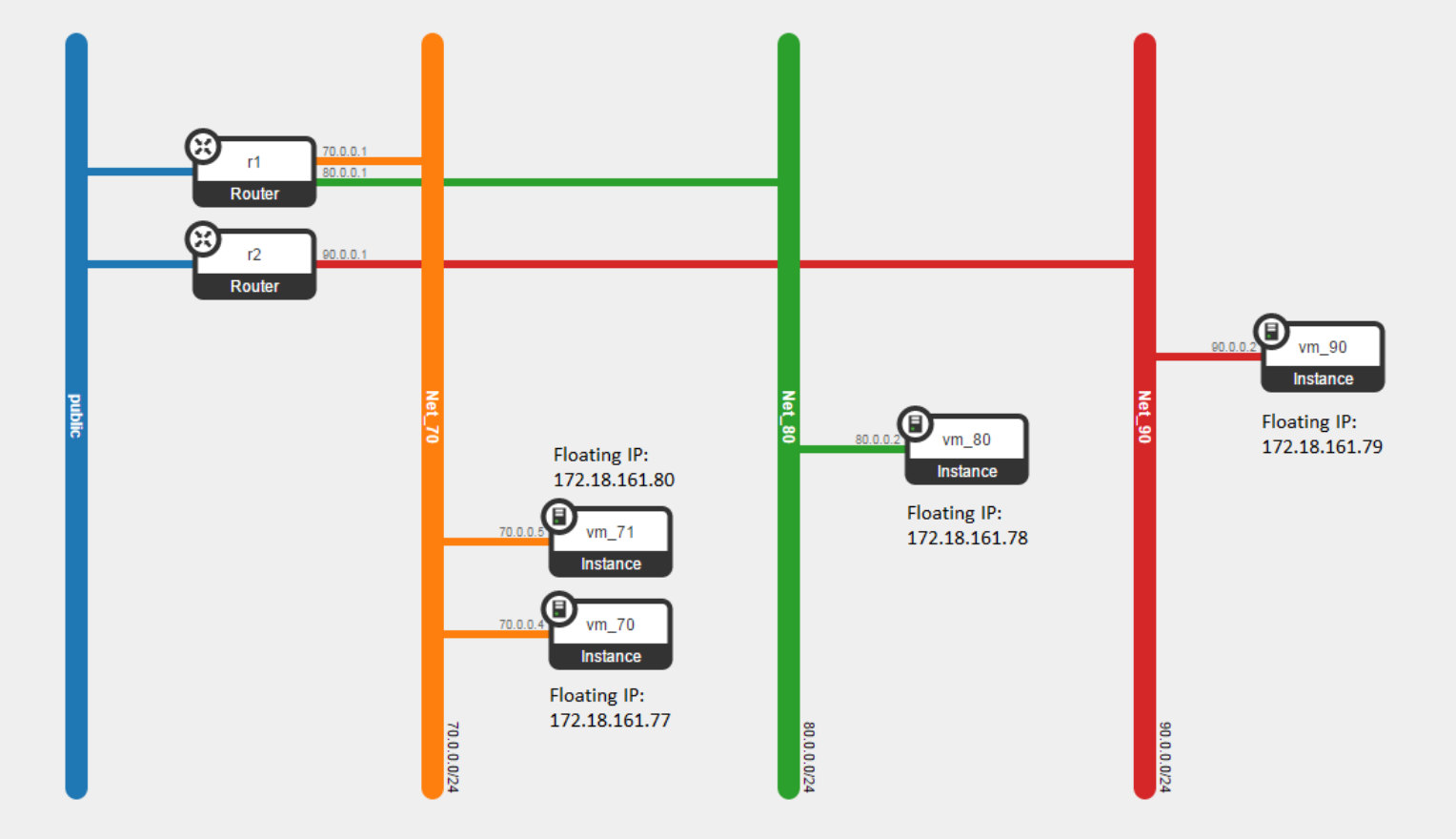
Configuring FWaaS service

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

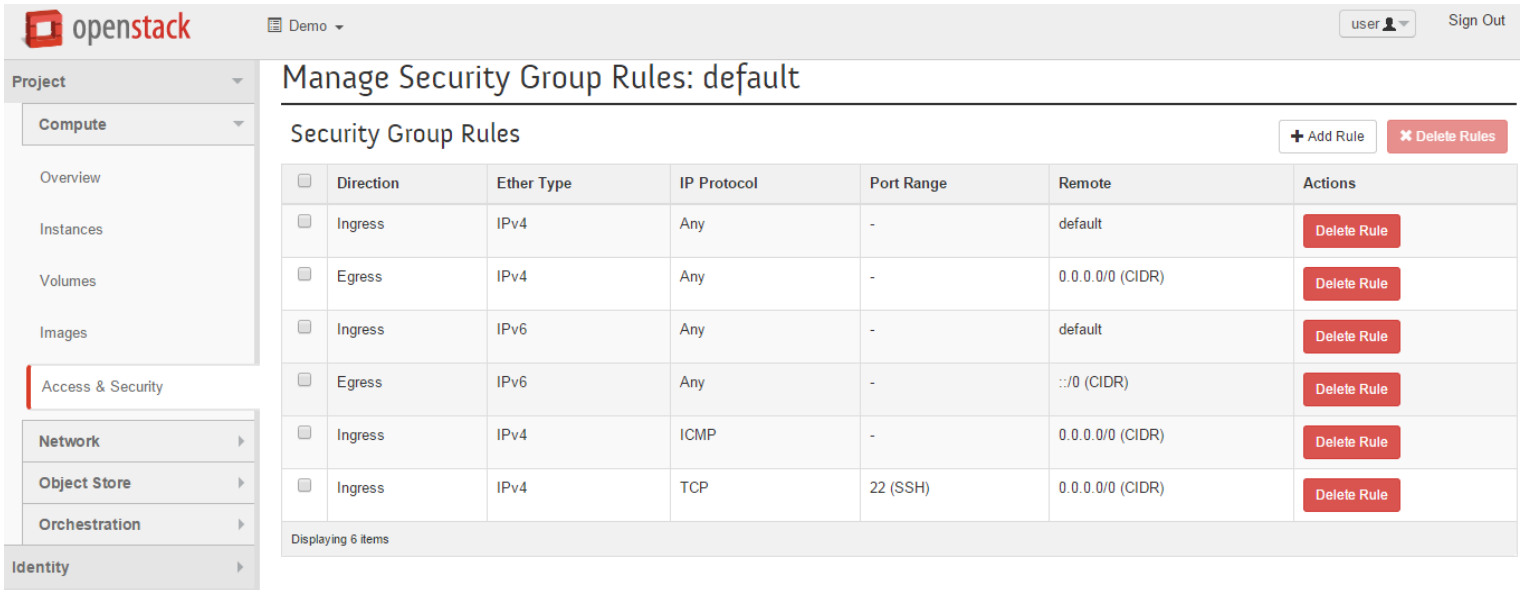
This section provides an example of configuration and step-by-step instructions for configuring the plugin.

Here is an example task:

We will have the following network architecture in our Project:



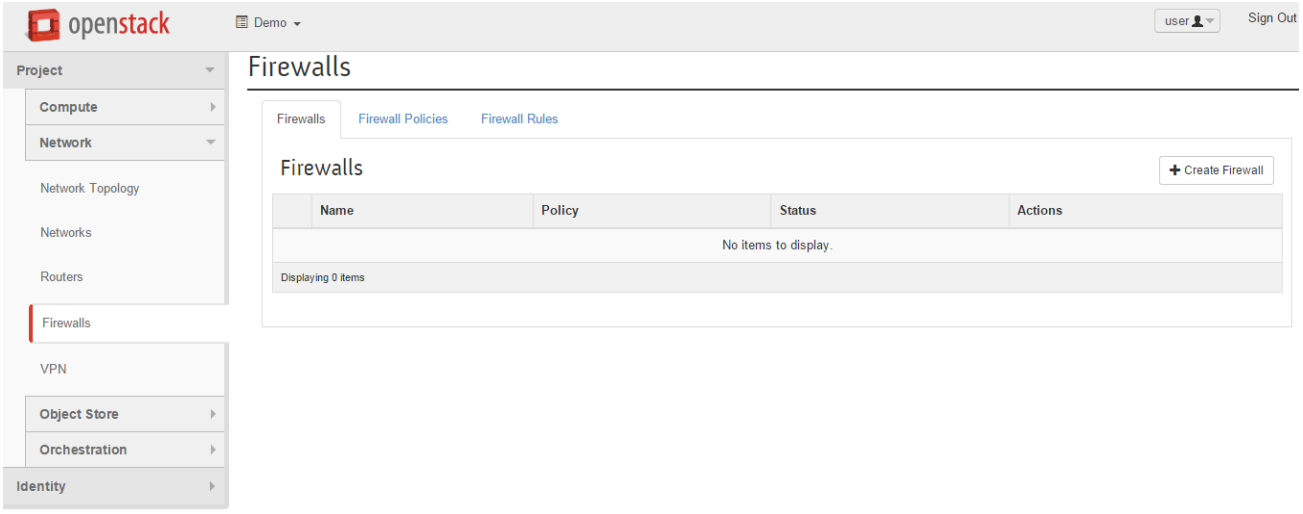
Before we start, we need to be remember that every Project in OpenStack is assigned the default security group for the cluster in its default form, which is usually restrictive. So you'll probably need to create a few additional rules in each Project's default security group: like a general ICMP rule,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



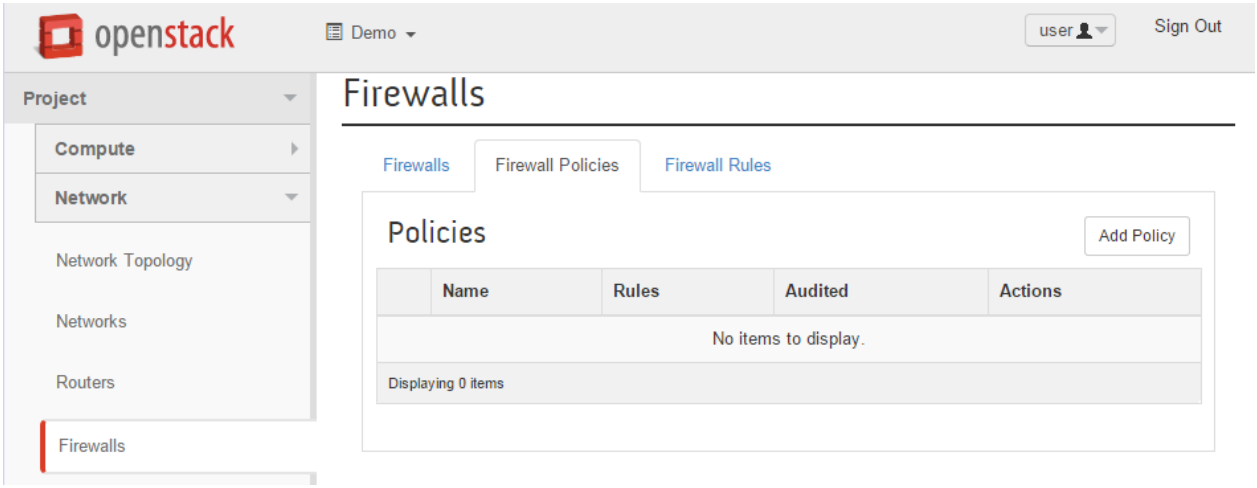
Let's get started with the testing of **connectivity between our VMs** (using **ping**). So, for the current state situation is the following (see the network topology above):

	VM_70		VM_71		VM_80		VM_90	
	Local IP 70.0.0.4	Floating IP 172.18.161.77	Local IP 70.0.0.5	Floating IP 172.18.161.80	Local IP 80.0.0.2	Floating IP 172.18.161.78	Local IP 90.0.0.2	Floating IP 172.18.161.79
VM_70	+	+	+	+	+	+	-	+
VM_71	+	+	+	+	+	+	-	+
VM_80	+	+	+	+	+	+	-	+
VM_90	-	+	-	+	-	+	+	+
My PC	-	+	-	+	-	+	-	+

1. Let's configure Firewall. To do that, please select *Network* option in the left-hand menu and click *Firewall*.

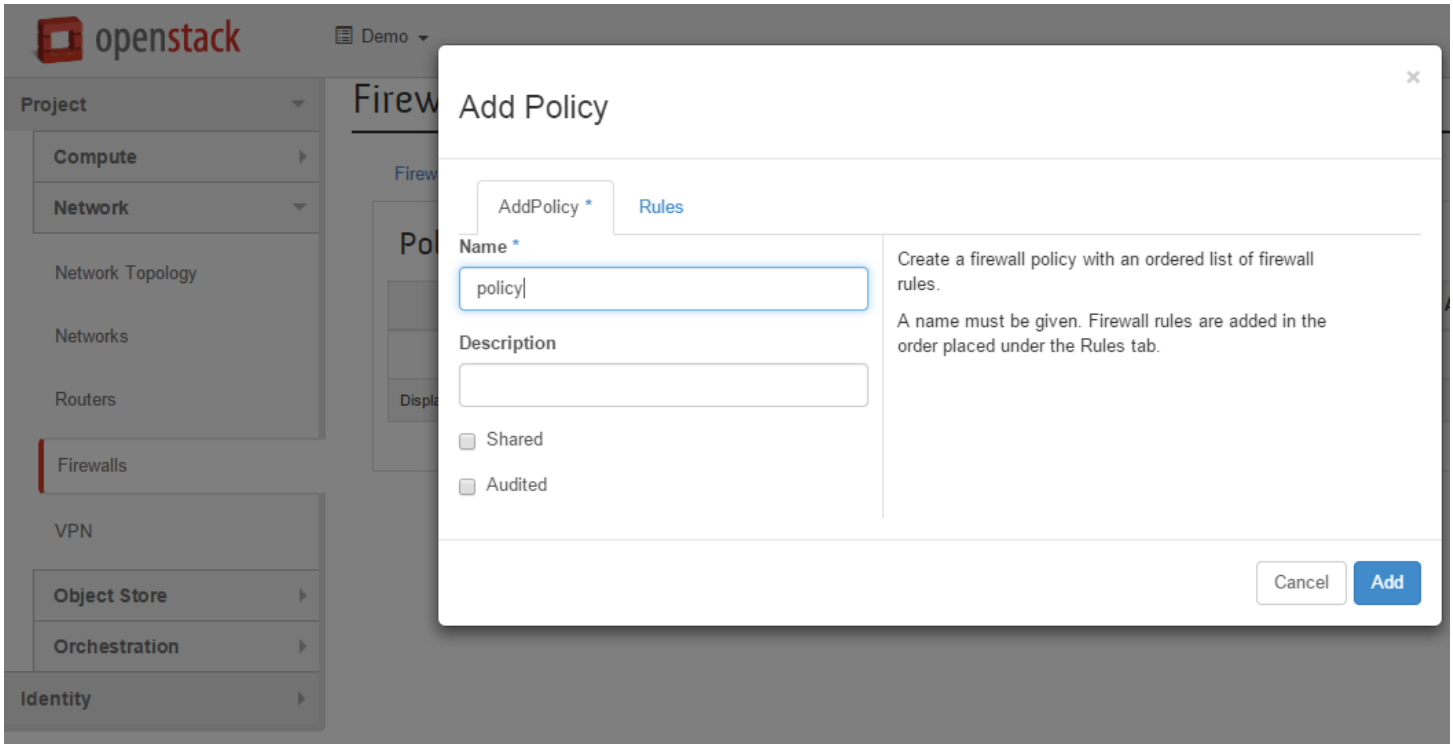


2. Create a **Policy.**
    a. Enter *Firewall Policies* tab and click *Add Policy* button (see the screenshot above).

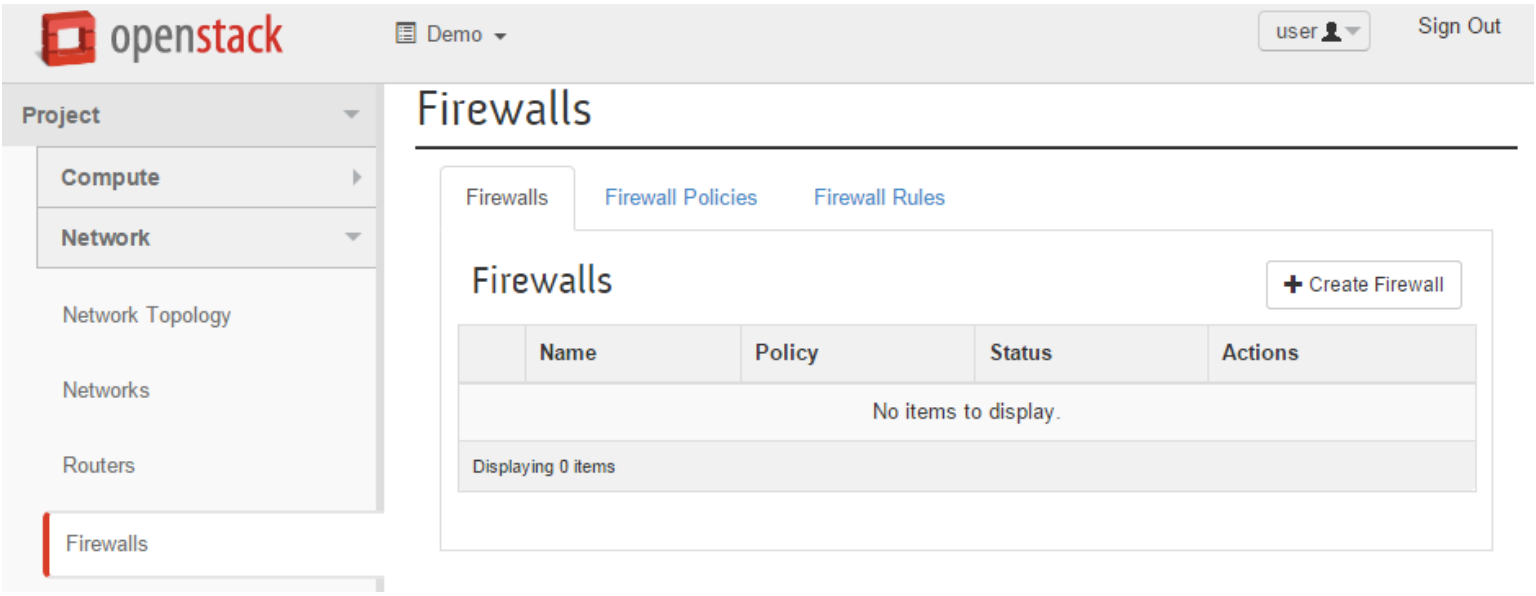


3. In *Add Policy* window, we should fill in policy name and description of this policy in the *Name* and *Description* fields. Also, here we can set *Shared* and *Audited* flags:
    - *Shared* - allow to share your policy with all other Projects.
    - *Audited* - indicate whether the particular firewall policy was audited or not by the creator of the firewall policy.

Click *Add* button to finish.

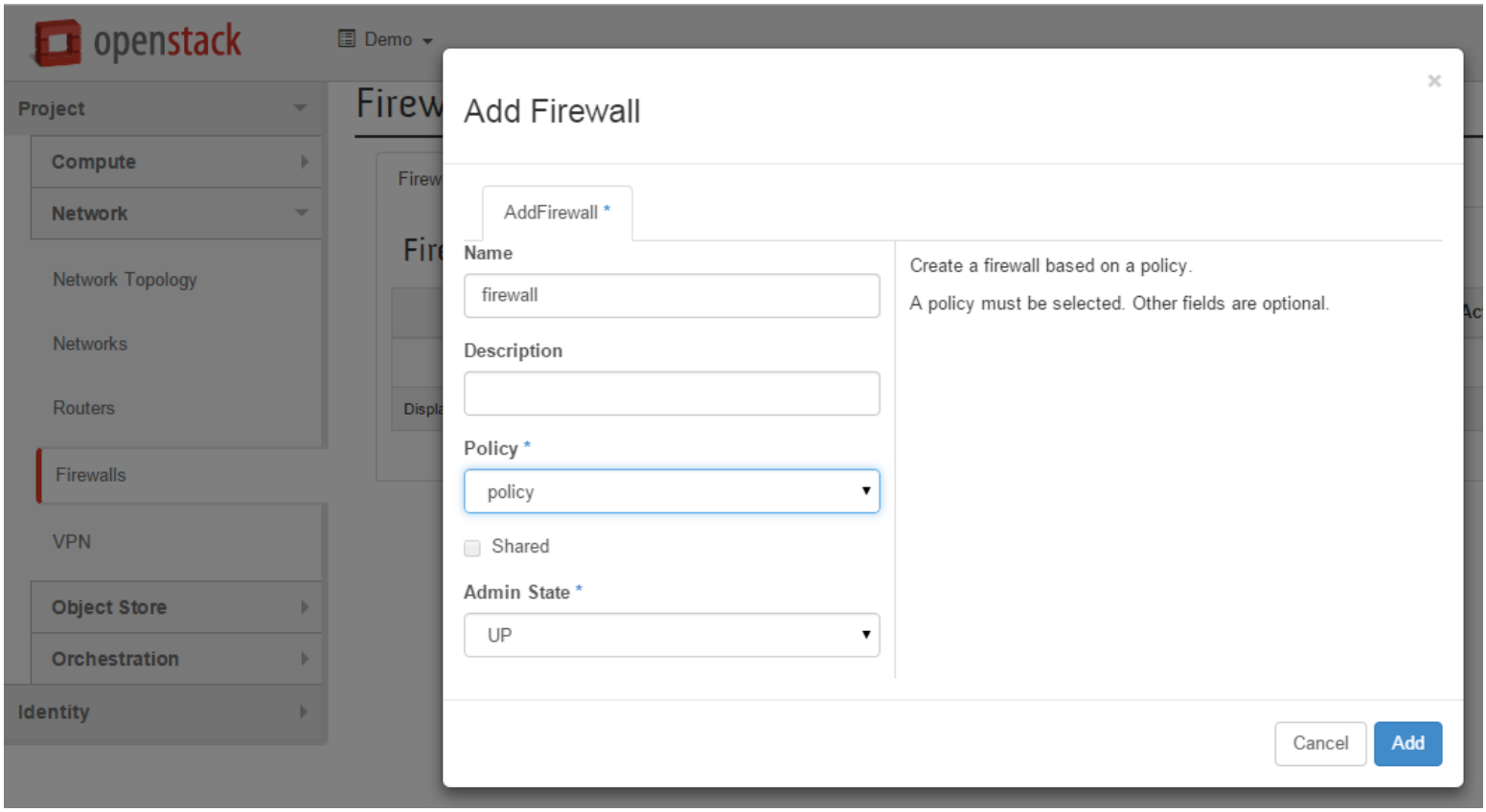


4. Create the **Firewall**.
    a. Enter *Firewalls* tab and click *Create Firewall* button. In the current implementation of the FWaaS plugin, we can create only one Firewall per Project and Firewall policy (rules) that is applied for all routers in this Project:



.

5. In *Add Firewall* window we should fill in *Name, Description* fields and choose our policy that was created in step 3.
    - *Admin State* option provide an ability to set UP or DOWN the Firewall.



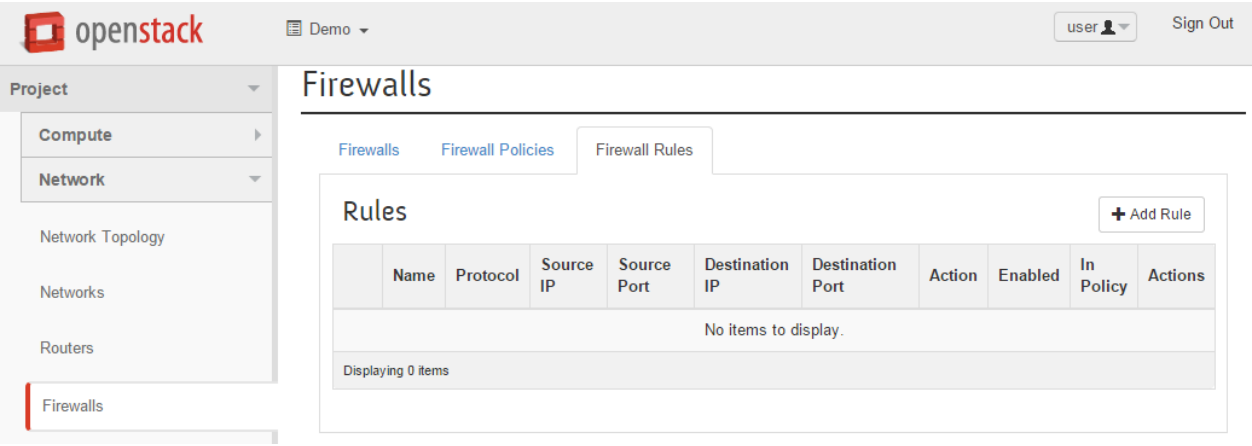
NOTE: The firewall remains in `PENDING_CREATE` state until you create a Networking router and attach an interface to it.

6. Let's test connectivity between our VMs one more time:

	VM_70		VM_71		VM_80		VM_90	
	Local IP 70.0.0.4	Floating IP 172.18.161.77	Local IP 70.0.0.5	Floating IP 172.18.161.80	Local IP 80.0.0.2	Floating IP 172.18.161.78	Local IP 90.0.0.2	Floating IP 172.18.161.79
VM_70	+	-	+	-	-	-	-	-
VM_71	+	-	+	-	-	-	-	-
VM_80	-	-	-	-	+	-	-	-
VM_90	-	-	-	-	-	-	+	-
My PC	-	-	-	-	-	-	-	-

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

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



8. 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 or TCP or UDP).
    - *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.

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

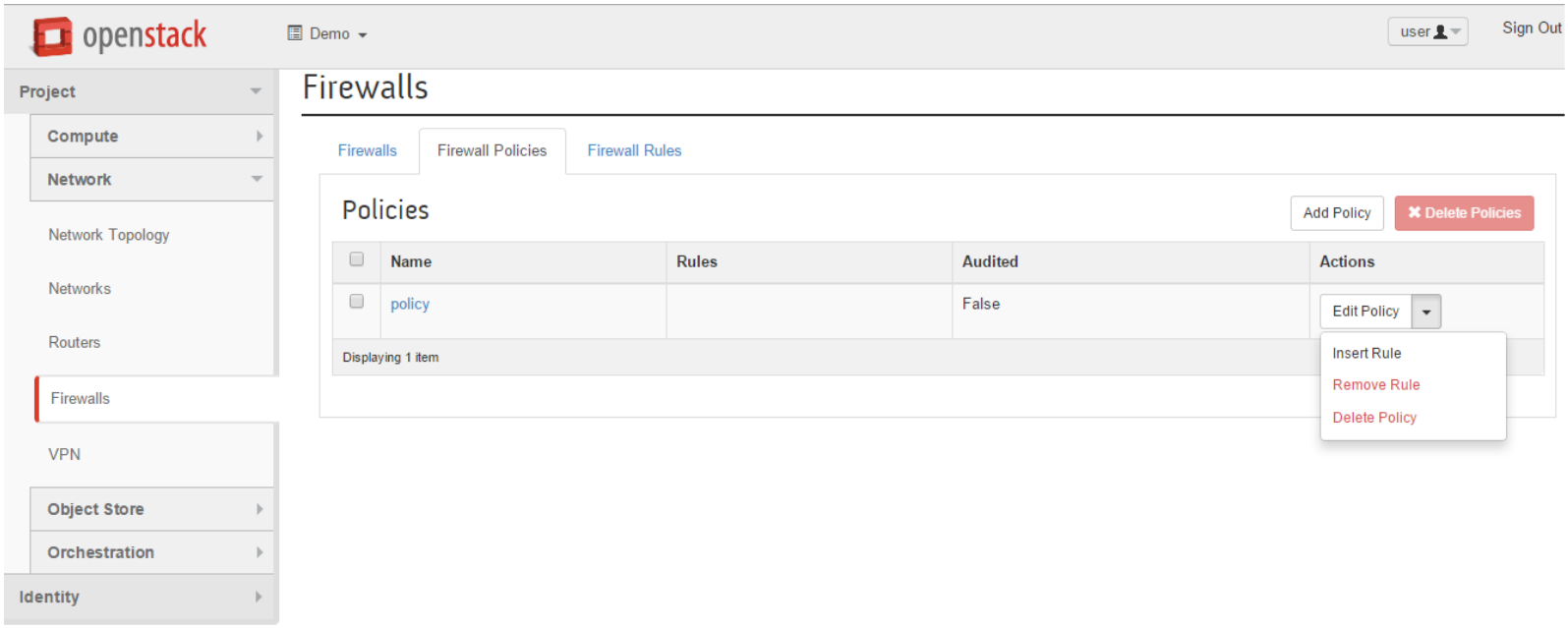
Enabled

Create a firewall rule.

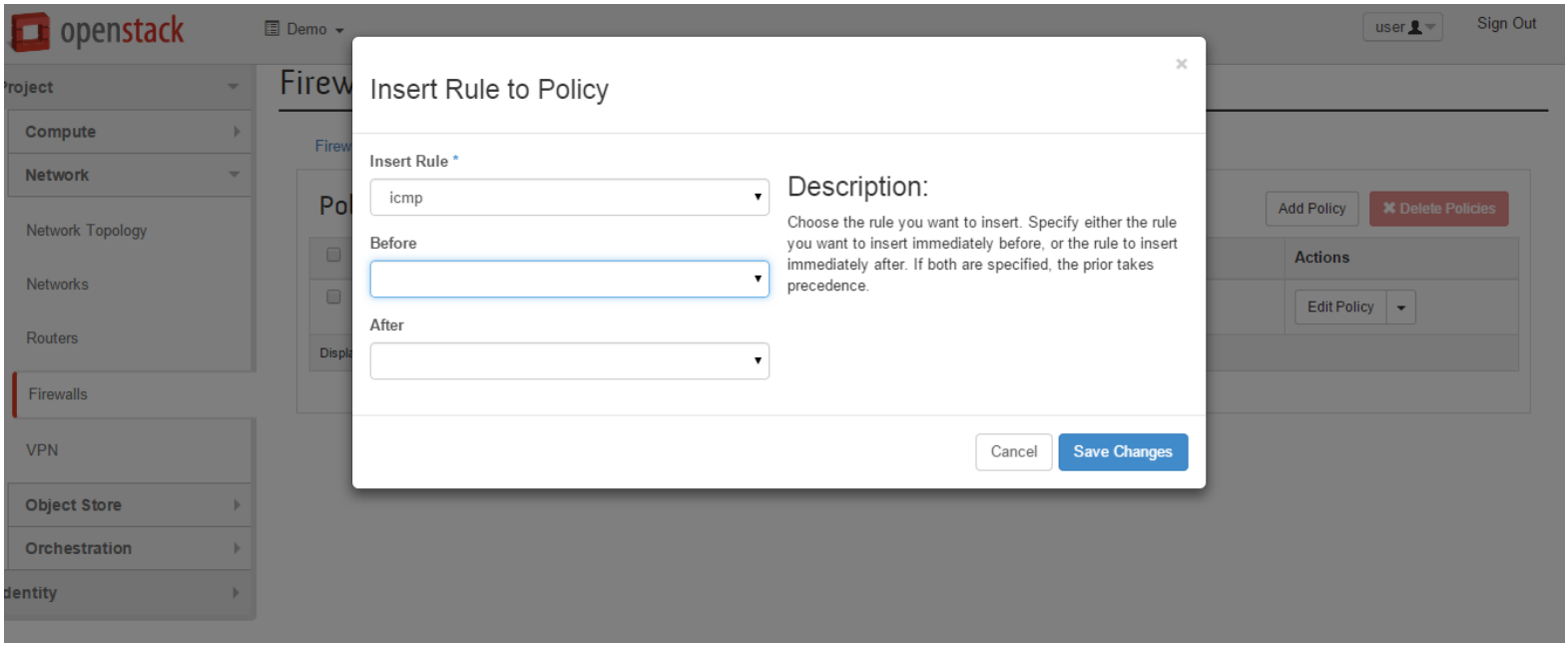
Protocol and action must be specified. Other fields are optional.

Cancel Add

9. Add the created rule into our policy.
    a. Enter *Firewall Policies*.
    b. In column for our policy, click drop-down button and select *Insert Rule*.



10. 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.



11. And let's test connectivity again:

	VM_70		VM_71		VM_80		VM_90	
	Local IP 70.0.0.4	Floating IP 172.18.161.77	Local IP 70.0.0.5	Floating IP 172.18.161.80	Local IP 80.0.0.2	Floating IP 172.18.161.78	Local IP 90.0.0.2	Floating IP 172.18.161.79
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.1

Appendix

#	Title of resource	Link on resource
1.	Firewall-as-a-Service Overview	Link