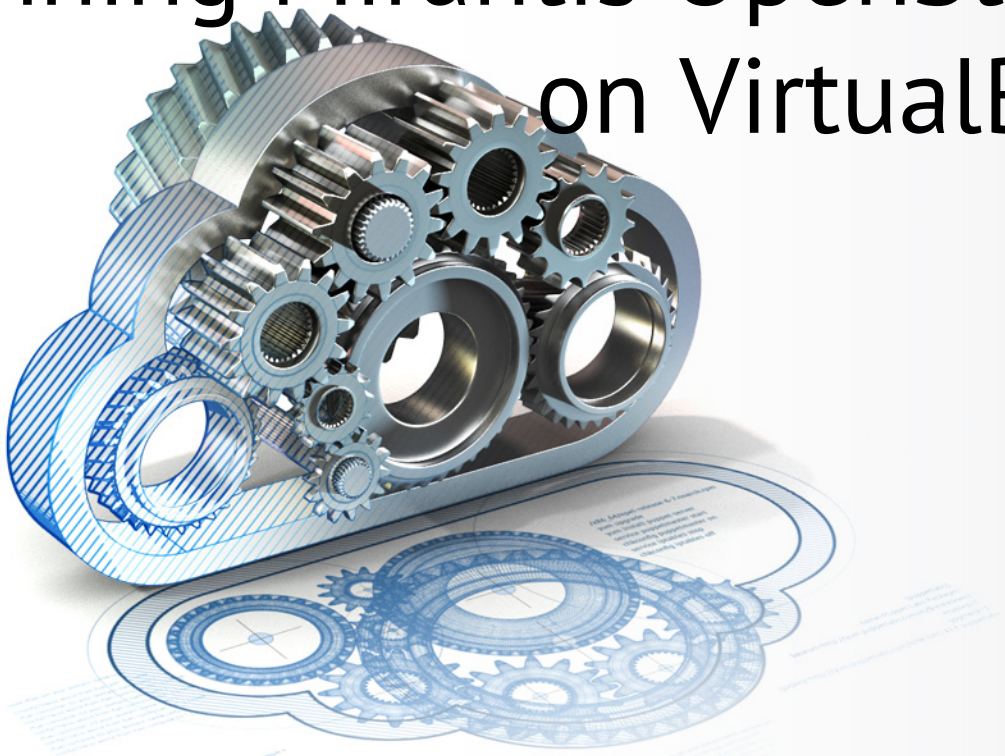
MIRANTIS

BUILT FOR openstack®

Mirantis OpenStack®

version 6.0

Running Mirantis OpenStack on VirtualBox



Contents

Preface

This documentation provides information on how to use Mirantis Fuel to deploy OpenStack environment. The information is for reference purposes and is subject to change.

Intended Audience

This documentation is intended for OpenStack administrators and developers; it assumes that you have experience with network and cloud concepts.

Documentation History

The following table lists the released revisions of this documentation:

Revision Date	Description
October, 2014	6.0 Technical Preview
December, 2014	6.0 GA

Introduction

You can install Fuel on VirtualBox and use that to deploy a Mirantis OpenStack environment for demonstration and evaluation purposes. Mirantis provides scripts that create and configure all the VMs required for a test environment, including the Master node and Slave nodes. See the Quick Start Guide for links and instructions.

This guide provides additional information about running Fuel and Mirantis OpenStack on VirtualBox.

The requirements for running Fuel on VirtualBox are:

A host machine with Linux, Windows or Mac OS. We recommend 64-bit host OS. The scripts have been tested on Mac OS 10.7.5, Mac OS 10.8.3, Ubuntu 12.04, Ubuntu 12.10, Fedora 19, OpenSUSE 12.2/12.3, and Windows 7 x64 + Cygwin_x64.

VirtualBox 4.2.16 (or later) is required, along with the extension pack. Both can be downloaded from http://www.virtualbox.org/.

Note

To run these scripts on Windows directly, you must first install Cygwin on your system; see the Cygwin installation page. You can also manually create the VMs to use for Fuel and the Slave nodes.

You need to install the **expect**, **openssh**, **ping** and **procps** packages, which do not install by default. Use the "-P expect,openssh,ping,procps" option to install these; for example:

```
setup-x86_64.exe -a x86_64  -P expect,openssh,ping,procps --quiet-mode \
   --site http://box-soft.com/
```

8 GB+ of RAM

Supports 4 VMs for Multi-node OpenStack installation (1 Master node, 1 Controller node, 1 Compute node, 1 Cinder node). The size of each VM should be reduced to 1536 MB RAM. For dedicated Cinder node, 768 MB of RAM is enough.

or

Supports 5 VMs for Multi-node with HA OpenStack installation (1 Master node, 3 combined Controller + Cinder nodes, 1 Compute node). The size of each VM should be reduced to 1280 MB RAM. This is less that the recommended amount of RAM amount per node for HA configurations (2048+ MB per controller) and may lead to unwanted issues.

Installing using automated scripts

When you unpack VirtualBox scripts, you will see the following important files and folders:

iso

Contains the ISO image used to install Fuel. You should download the ISO from the portal to this directory or copy it into this directory after it is downloaded. If this directory contains more than one ISO file, the installation script uses the most recent one.

config.sh

Configuration file that allows you to specify parameters that automate the Fuel installation. For example, you can select how many virtual nodes to launch, as well as how much memory, disk, and processing to allocate for each.

launch.sh

This is the script you run to install Fuel. It uses the ISO image from the `iso` directory, creates a VM, mounts the image, and automatically installs the Fuel Master node. After installing the Master node, the script creates Slave nodes for OpenStack and boots them via PXE from the Master node. When Fuel is installed, the script gives you the IP address to use to access the Web-based UI for Fuel. Use this address to deploy your OpenStack environment.

Manual Installation

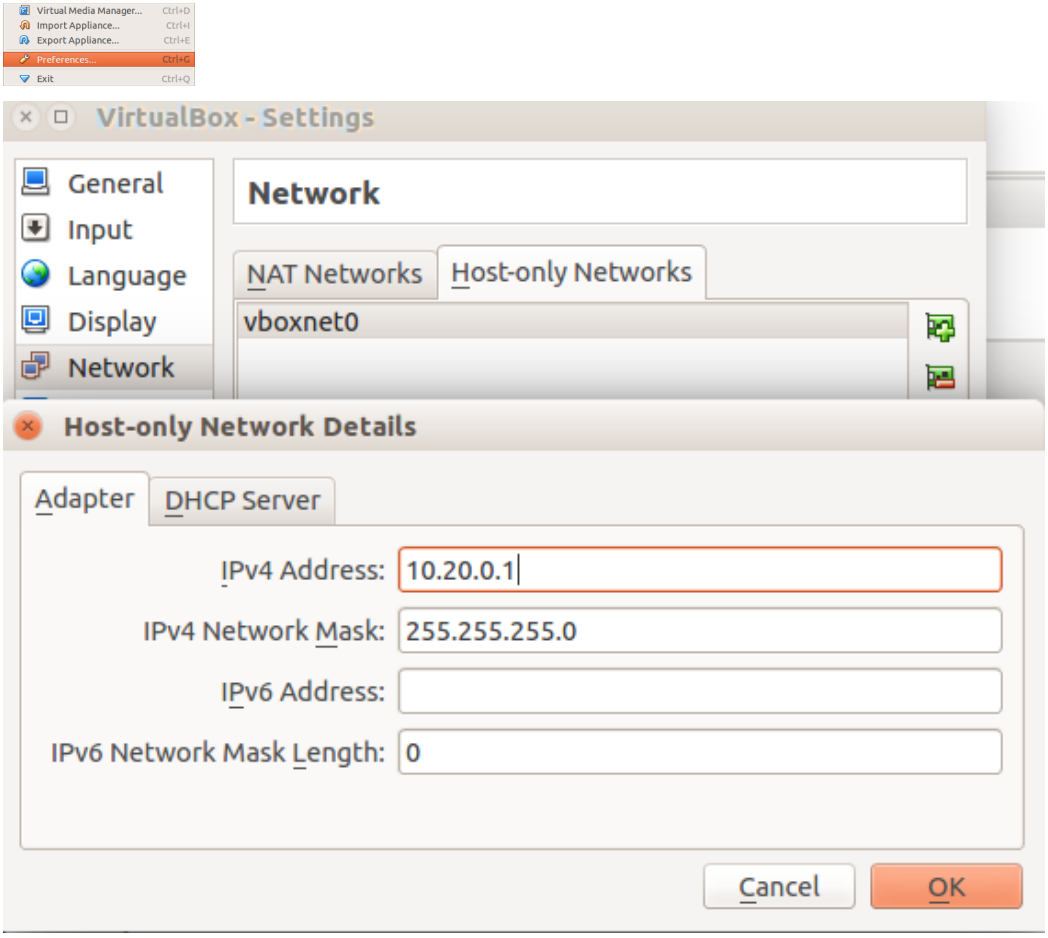
> ***Note***
>
> The following steps are suitable only for setting up a vanilla OpenStack environment for evaluation purposes only.

If you cannot or would rather not run our helper scripts, you can still run Fuel on VirtualBox by following these steps.

Deploying the Master Node Manually

First, create the Master node VM.

1. Configure the host-only interface vboxnet0 in VirtualBox by going to *File* -> *Preferences* -> *Network*, then on the *Host-only Networks* tab click the screwdriver icon.



- IP address: 10.20.0.1
- Network mask: 255.255.255.0
- DHCP Server: disabled

2. Create a VM for the Fuel Master node with the following parameters:
- OS Type: Linux
- Version: Ubuntu (64bit)
- RAM: 1536+ MB (2048+ MB recommended)
- HDD: 50 GB with dynamic disk expansion

3. Modify your VM settings:
- Network: Attach *Adapter 1* to *Host-only adapter vboxnet0*

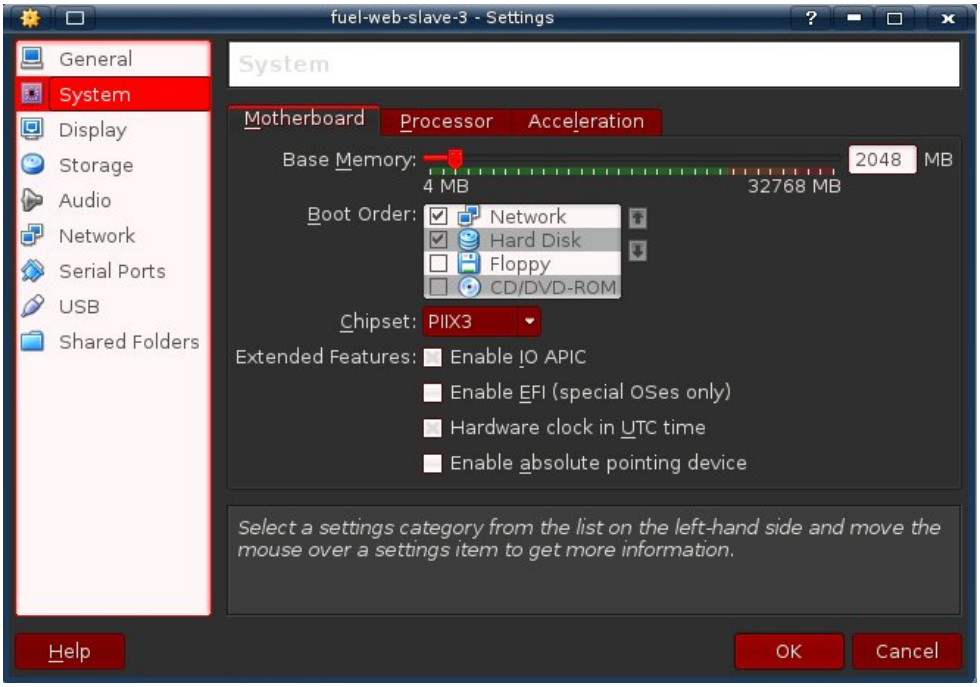
4. Power on the VM in order to start the installation. Choose your Fuel ISO when prompted to select start-up disk.
5. Wait for the Welcome message with all information needed to login into the UI of Fuel.

Adding Slave Nodes Manually

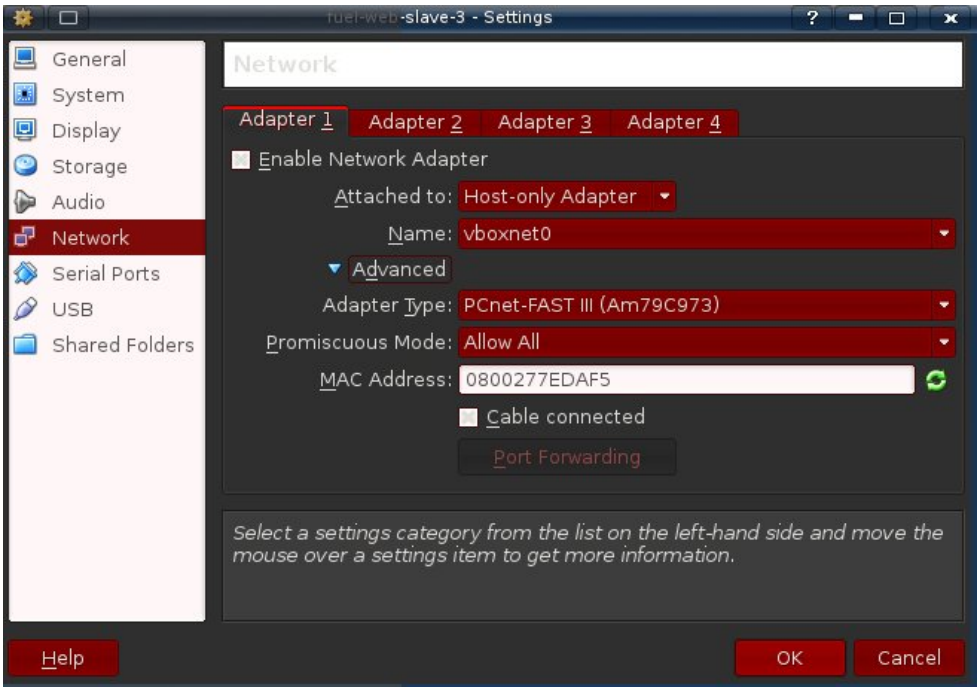
Next, create Slave nodes where OpenStack needs to be installed.

1. Create 3 or 4 additional VMs with the following parameters:
- OS Type: Linux, Version: Ubuntu (64bit)
- RAM: 1536+ MB (2048+ MB recommended)
- HDD: 50+ GB, with dynamic disk expansion
- Network 1: host-only interface vboxnet0, PCnet-FAST III device

2. Set Network as first in the boot order:



3. Configure two or more network adapters on each VM (in order to use single network adapter for each VM you should choose "Use VLAN Tagging" later in the Fuel UI):



4. Open "advanced" collapse, and check following options:

- Set Promiscuous mode to "Allow All"

- Set Adapter type to "PCnet-FAST III"
- Set Cable connected to On

Networking

By default, the launch script for the VirtualBox deployment creates three host-interface adapters. Basically, networking works as if you have 3 switches, one of which is connected to a VM network interface. This means that you have L2 connectivity between VMs on interfaces with the same name. If, for example, you try to move the management network to *eth1* on the Controller node, and the same network to *eth2* on the Compute node, then there will be no connectivity between OpenStack services, despite being configured to exist on the same VLAN. You can validate your network settings prior to deployment by clicking the "Verify Networks" button. If you need to access the OpenStack REST API over the Public network, VNC console of VMs, Horizon in HA mode or VMs, refer to this section: *Deployment configuration to access OpenStack API and VMs from host machine*.

Deployment configuration to access OpenStack API and VMs from host machine

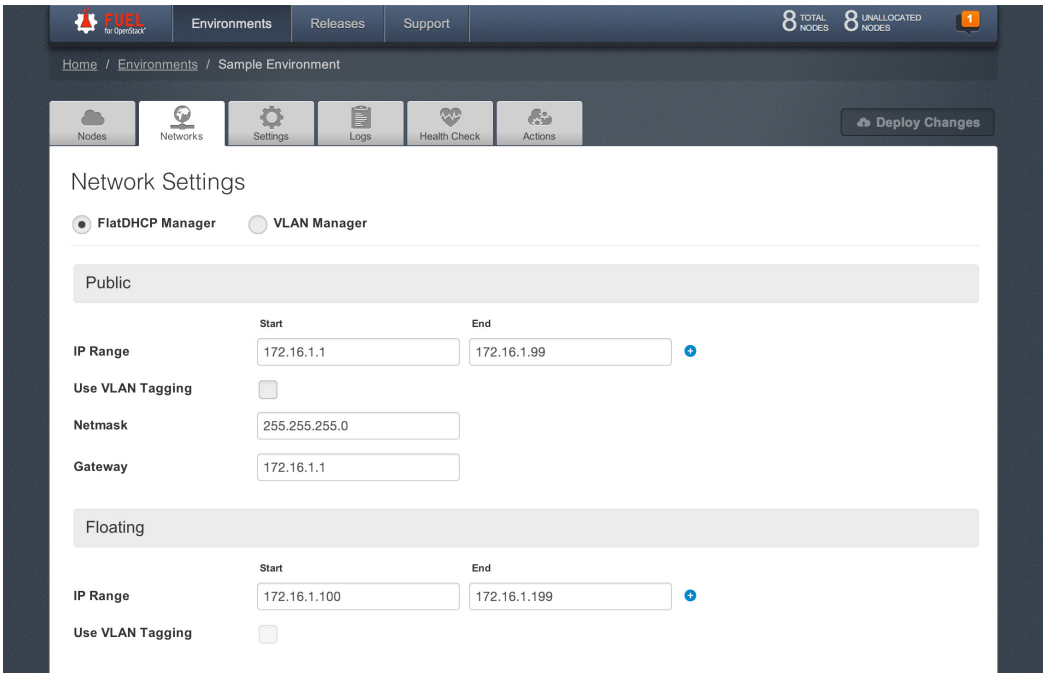
Follow the instructions in *Create a new OpenStack environment* and *Configure your Environment* to create and configure your OpenStack environment. Most of the steps are the same for a VirtualBox deployment and the bare-metal deployment. The one exception is networking, where the VirtualBox deployment requires some different settings.

Helper scripts for VirtualBox create the network adapters *eth0*, *eth1*, *eth2* which are represented on the host machine as *vboxnet0*, *vboxnet1*, *vboxnet2* correspondingly, and assigned IP addresses for adapters:

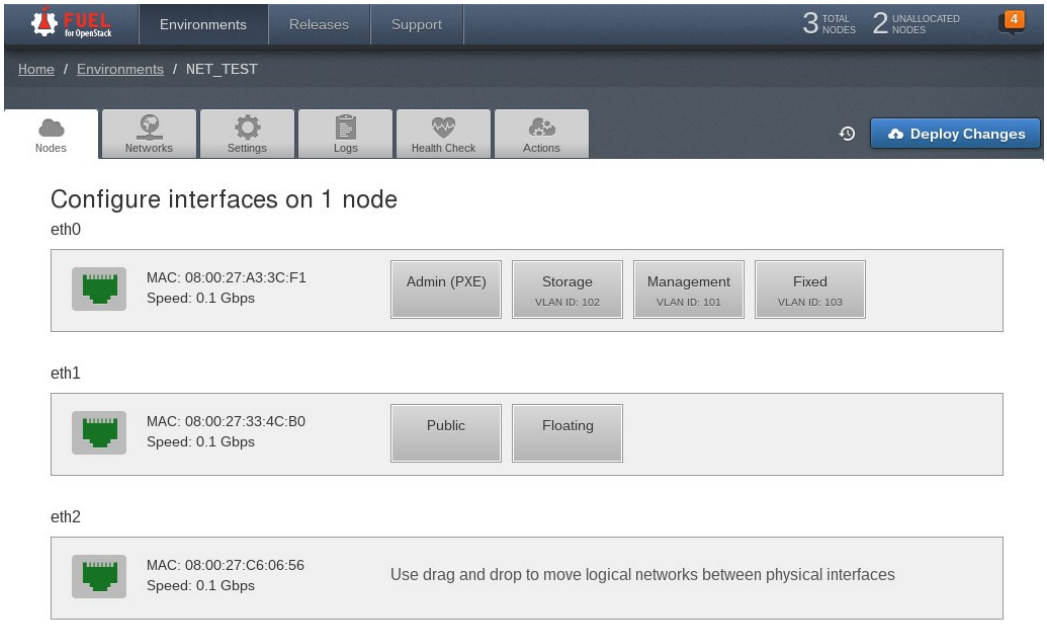
```
vboxnet0 - 10.20.0.1/24
vboxnet1 - 172.16.0.1/24
vboxnet2 - 172.16.1.1/24
```

For the demo environment on VirtualBox, the first network adapter is used to run Fuel network traffic, including PXE discovery.

To access the Horizon and OpenStack RESTful API via the Public *logical network* from the host machine, you must have a route from your host to the Public IP address on the OpenStack Controller. Also, if you need to access a VM's Floating IP, you must also have a route to the Floating IP on the Compute node, which is bound to the Public interface there. To make this configuration possible on the VirtualBox demo environment, you must run the Public network untagged. On the image below, you can see the configuration of Public and Floating networks which allows this:



By default, Public and Floating networks run on the first network interface. This must be changed on each node, to run these networks on eth1 by setting the configuration as shown here:



If you use the default configuration in VirtualBox scripts, and use the settings shown on the images above, you should be able to access OpenStack Horizon via the Public network after the installation.

If you want to enable Internet access on VMs that are provisioned by OpenStack, you must configure NAT on the host machine. When packets reach the *vboxnet1* interface, according to the OpenStack settings tab, they must know the way out of the host. For Ubuntu, the following command, executed on the host, makes this happen:

```
sudo iptables -t nat -A POSTROUTING -s 172.16.1.0/24 \! -d 172.16.1.0/24 -j MASQUERADE
```

To access VMs managed by OpenStack, you must provide IP addresses from the Floating IP range. When the OpenStack environment is deployed and VM is provisioned there, you have to associate one of the Floating IP addresses from the pool with this VM, whether in Horizon or via Nova CLI. By default, OpenStack blocks all the traffic to the VM. To allow the connectivity to the VM, you need to configure *security groups*. This can be done in Horizon or from the OpenStack Controller. For example, the following commands issued from the OpenStack controller allow ICMP and SSH traffic to pass on to the VM:

```
. /root/openrc
nova secgroup-add-rule default icmp -1 -1 0.0.0.0/0
nova secgroup-add-rule default tcp 22 22 0.0.0.0/0
```

IP ranges for Public and Management networks (172.16.*.*) are defined in the `config.sh` script. If default values do not fit your needs, you are free to change them, but you must make the modifications before running the **launch.sh** command to install the Fuel Master node.

Additional Notes

- Do not run VirtualBox as the root user or as any user with superuser permissions. You must run it as a normal user and add this user name to the *vboxusers* security group, which is automatically created as part of the VirtualBox installation. The following command adds the "myname" user to the vboxusers group:

  ```
  sudo useradd -G vboxusers myname
  ```

- If the target server where you run VirtualBox is not running X11, you must modify the scripts to use the headless option: "VBoxManage startvm ... --type headless"