US HEADQUARTERS

525 Almanor Avenue, 4th Floor

Sunnyvale, CA 94085

+1-650-963-9828 Phone

+1-650-968-2997 Fax

Run book for OpenStack

Deployment guide

Cluster CUSTOMER ENV

OpenStack RELEASE

*Version 1*

*DATE*

*Author(s) <Nikita Koshikov, koshikov@mirantis.com>*

Table of contents

[Document purpose](#h.3dy6vkm)

[Key terms, acronyms and abbreviations](#h.1t3h5sf)

[Introduction](#h.4d34og8)

[Types of Nodes](#h.2s8eyo1)

[Architecture](#h.17dp8vu)

[Network Layout](#h.3rdcrjn)

[Nodes](#h.26in1rg)

[Access Information](#h.lnxbz9)

[Fuel Master Node Installation](#h.35nkun2)

[Post installation customization](#h.1ksv4uv)

[OpenStack Environment Deployment](#h.44sinio)

[Creation of environment:](#h.2jxsxqh)

[Network settings:](#h.z337ya)

[Environment settings:](#h.3j2qqm3)

[Nodes network allocation:](#h.1y810tw)

[Controller nodes HDD allocation:](#h.4i7ojhp)

[Compute nodes HDD allocation:](#h.2xcytpi)

[Storage nodes HDD allocation:](#h.1ci93xb)

[Pre Deployment Customization](#h.3whwml4)

[Adding and Removing Compute & Storage Nodes](#h.2bn6wsx)

[Post Deployment Customization](#h.qsh70q)

[[Extensions implemented in OpenStack to define the provider network]](#h.3as4poj)

[[Creating provider networks within OpenStack]](#h.1pxezwc)

[Cloud Verification Test Results](#h.49x2ik5)

[Cli and Web UI verification results](#h.2p2csry)

# Document purpose

This document describes the installation process of Customer’s OpenStack cloud.

# Key terms, acronyms and abbreviations

|  |  |
| --- | --- |
| **Term/acronym/abbreviation** | **Definition** |
| VM | Virtual Machine |
| CLI | Command-line user interface |
| UI | Graphical user interface |

# Introduction

This document describes the installation process of Customer’s staging OpenStack cloud. For more detailed information use official Mirantis OpenStack documentation.

<https://docs.mirantis.com/openstack/fuel/fuel-7.0/>

# Types of Nodes

* Controller:  
  The controller node hosts all OpenStack management services. It runs MySQL and RabbitMQ used for communication between nodes. The Glance image service and the Ceph Monitor service also run there.
* Compute:  
  Compute nodes are responsible for running VMs.
* Storage

Storage nodes are storing Glance images, volumes, VM’s block devices. The Ceph OSD Daemon is running there.

* Fuel master:  
  Fuel master node is responsible for managing OpenStack installation by UI or CLI, collecting log files.
* Mongo

Ceilometer uses mongo database as a backend for metrics storing.

* Elasticsearch\_kibana, Influxdb\_grafana and Infrastructure\_alerting

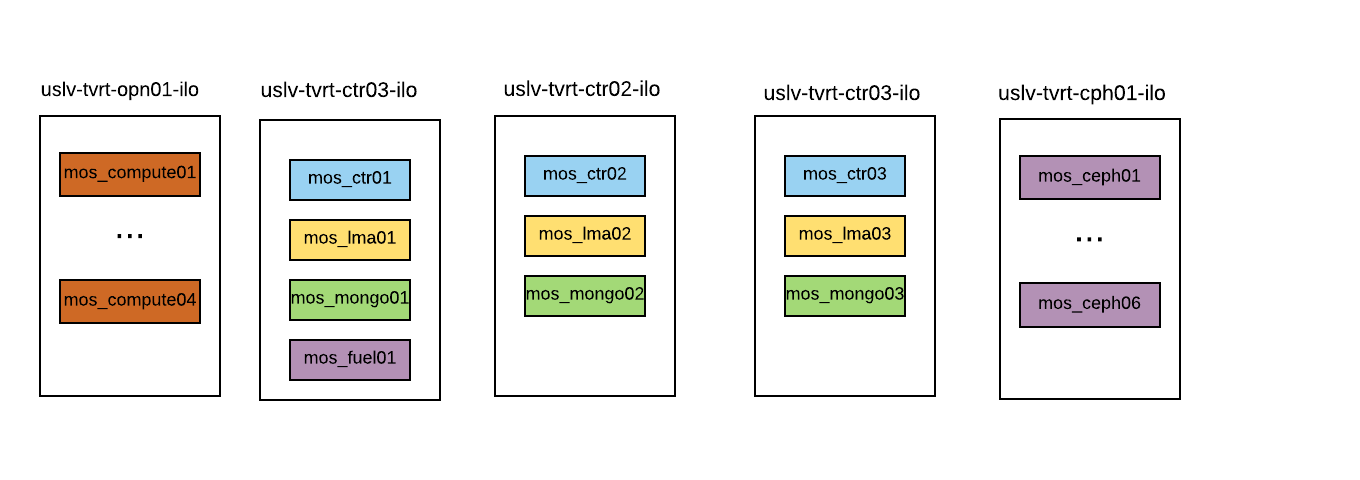
These nodes are used as a monitoring solution that stores and collect data, required for notifications and alerting support.

# Architecture

Staging environment build on top of VMs, that hosted on 5 hardware nodes. Each node is Supermicro server with:

* Intel(R) Xeon(R) CPU E5-2650 v2 @ 2.60GHz
* 256G Ram
* 12x4Tb disk without RAID

Logically this looks like scheme below:



# Prepare hardware nodes:

All 5 servers were provisioned manually, using remote console. Ubuntu 14.04 was installed.

These are steps that were done in order to run VMs on top of hardware:

* Setup networking(adapt IPs per node)

|  |
| --- |
| # cat /etc/network/interfaces  auto lo  iface lo inet loopback  iface em3 inet manual  iface p2p1 inet manual  iface p2p2 inet manual  auto br-admin  iface br-admin inet static  bridge\_ports em3  address 10.200.160.22  netmask 255.255.255.0  gateway 10.200.160.1  dns-nameservers 10.200.13.249 10.200.22.249 10.200.7.249  auto br-vlans  iface br-vlans inet manual  bridge\_ports p2p1  auto br-provider  iface br-provider inet manual  bridge\_ports p2p2 |

* Install KVM, using [this](https://help.ubuntu.com/community/KVM/Installation) guide.
* Create LVM(adapt commands per server)

|  |
| --- |
| # pvcreate /dev/sdb  # vgcreate MOS /dev/sub  # lvcreate --name VM\_Images --size 10.9T MOS  # vim /etc/fstab  /dev/mapper/MOS-VM\_Images /var/lib/libvirt/images ext4 defaults 0 2  # mkfs.ext4 /dev/MOS/VM\_Images  # mount /dev/MOS/VM\_Images  # virsh pool-define-as MOS\_VM\_Images dir - - - - "/var/lib/libvirt/images"  # virsh pool-autostart MOS\_VM\_Images  # virsh pool-start MOS\_VM\_Images |

* Create VM disks and define libvirt-xml VMs (adapt per server)

|  |
| --- |
| mkdir VMs  #copy libvirt templates to VMs  grep 'source file' \*.xml  sudo qemu-img create -f qcow2 /var/lib/libvirt/images/mos\_ctr01\_vda.img 100G  sudo qemu-img create -f qcow2 /var/lib/libvirt/images/mos\_ctr01\_vdb.img 1T  sudo qemu-img create -f qcow2 /var/lib/libvirt/images/mos\_lma01\_vda.img 100G  sudo qemu-img create -f qcow2 /var/lib/libvirt/images/mos\_lma01\_vdb.img 1T  sudo qemu-img create -f qcow2 /var/lib/libvirt/images/mos\_mongo01\_vda.img 100G  sudo qemu-img create -f qcow2 /var/lib/libvirt/images/mos\_mongo01\_vdb.img 1T  sudo qemu-img create -f qcow2 /var/lib/libvirt/images/mos\_fuel01\_vda.img 1T  virsh start mos\_fuel01  virsh autostart mos\_fuel01  virsh define mos\_fuel01  virsh define mos\_ctr01.xml  virsh define mos\_lma01.xml  virsh define mos\_mongo01.xml  virsh start mos\_ctr01  virsh start mos\_lma01  virsh start mos\_mongo01  virsh autostart mos\_ctr01  virsh autostart mos\_lma01  virsh autostart mos\_mongo01 |

# Network Layout

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Network name** | **Speed** | **Port mode** | **IP Range** | **VLAN** | **Interface** |
| IPMI network | 1 Gbps | Untagged | 10.200.30/24 | - | - |
| Admin/PXE network | 1 Gbps | Untagged | 10.200.160.0/24 | 176 | em1 |
| Management network | 10 Gbps | Tagged | 10.200.153.0/24 | 172 | br-vlans |
| Storage network | 10 Gbps | Tagged | 10.200.154.0/24 | 173 | br-vlans |
| Public network | 10 Gbps | Tagged | 10.200.155.0/24 | 174 | br-vlans |
| Provider network | 10 Gbps | Tagged | 172.21.18.224/28 | 996 | br-provider |

# Nodes(hardware + access)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Hostname** | **ILO IP address** | **Admin network IP address** | **User** | **Password** |
| uslv-tvrt-opn01-ilo | 10.200.30.187 | 10.200.160.21 | mirantis | r00tme |
| uslv-tvrt-ctr03-ilo | 10.200.30.188 | 10.200.160.22 | mirantis | r00tme |
| uslv-tvrt-ctr02-ilo | 10.200.30.190 | 10.200.160.24 | mirantis | r00tme |
| uslv-tvrt-ctr01-ilo | 10.200.30.189 | 10.200.160.23 | mirantis | r00tme |
| uslv-tvrt-cph01-ilo | 10.200.30.191 | 10.200.160.25 | mirantis | r00tme |
| fuel.domain.tld | - | 10.200.160.30 | root | r00tme |

# Nodes(VMs)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Hostname** | **Role** | **Admin network IP address** | **CPUxCores** | **RAM** | **HDD** |
| node-26.domain.tld | controller | 10.200.160.48 | 8 | 64GB | 1x100G+1x1TB |
| node-27.domain.tld | controller | 10.200.160.49 | 8 | 64GB | 1x100G+1x1TB |
| node-34.domain.tld | controller | 10.200.160.51 | 8 | 64GB | 1x100G+1x1TB |
| node-23.domain.tld | mongo | 10.200.160.47 | 4 | 128GB | 1x100G+1x1TB |
| node-23.domain.tld | mongo | 10.200.160.45 | 4 | 128GB | 1x100G+1x1TB |
| node-23.domain.tld | mongo | 10.200.160.46 | 4 | 128GB | 1x100G+1x1TB |
| node-25.domain.tld | ceph-osd | 10.200.160.42 | 4 | 32GB | 1x100G+4x350Gb |
| node-32.domain.tld | ceph-osd | 10.200.160.52 | 4 | 32GB | 1x100G+4x350Gb |
| node-38.domain.tld | ceph-osd | 10.200.160.54 | 4 | 32GB | 1x100G+4x350Gb |
| node-36.domain.tld | ceph-osd | 10.200.160.55 | 4 | 32GB | 1x100G+4x350Gb |
| node-35.domain.tld | ceph-osd | 10.200.160.53 | 4 | 32GB | 1x100G+4x350Gb |
| node-37.domain.tld | ceph-osd | 10.200.160.41 | 4 | 32GB | 1x100G+4x350Gb |
| node-7.domain.tld | elasticsearch | 10.200.160.37 | 8 | 32GB | 1x100G+1x1Tb |
| node-8.domain.tld | influxdb | 10.200.160.38 | 8 | 32GB | 1x100G+1x1Tb |
| node-9.domain.tld | alerting | 10.200.160.39 | 8 | 32GB | 1x100G+1x1Tb |
| node-24.domain.tld | compute | 10.200.160.43 | 8 | 60GB | 1x100G+1x1Tb |
| node-28.domain.tld | compute | 10.200.160.44 | 8 | 60GB | 1x100G+1x1Tb |
| node-30.domain.tld | compute | 10.200.160.40 | 8 | 60GB | 1x100G+1x1Tb |
| node-31.domain.tld | compute | 10.200.160.50 | 8 | 60GB | 1x100G+1x1Tb |

# Access Information

|  |  |  |
| --- | --- | --- |
| **Host** | **IP address** | **Comment** |
| Fuel UI master node URL | <http://10.200.160.30:8000> |  |
| Fuel UI credentials | admin / <admin> | Please see [here](https://docs.mirantis.com/openstack/fuel/fuel-7.0/user-guide.html#change-fuel-passwd-ug)[[1]](#footnote-1) how to change Fuel password |
| Fuel Master node IP address | 10.200.160.30 |  |
| Fuel SSH credentials (console) | root / <r00tme> |  |
| OpenStack nodes SSH credentials (console) |  | By default password based authentication disabled for OpenStack boxes, ssh to OpenStack nodes possible only from Fuel master nodes which has ssh access key for slave nodes. Passwords can be changed from consoles. More information can be found at Fuel Operations Guide [here](https://docs.mirantis.com/openstack/fuel/fuel-8.0/pdf/Mirantis-OpenStack-8.0-OperationsGuide.pdf)[[2]](#footnote-2) . |
|  |  |  |
| OpenStack Horizon URL | <http://10.200.143.22> |  |
| OpenStack credentials | admin / <admin> |  |

# Fuel Master Node Installation

Step by step manual for installation of Fuel master node available on the Mirantis website [here](https://docs.mirantis.com/openstack/fuel/fuel-8.0/fuel-install-guide.html#install-install-fuel-master-node)[[3]](#footnote-3) .

Following settings were used during installation of Fuel node:

|  |  |
| --- | --- |
| Hostname | fuel.domain.tld |
| Enabled Interface | eth0 |
| Interface for PXE | eth0 |
| IP address | 10.200.160.30 |
| Management Interface | eth0 |
| Gateway | 10.200.160.1 |
| Network mask | 255.255.255.0 |
| DHCP Pool range | 10.200.160.31 - 10.200.160.254 |
| Domain | domain.tld |
| Search domain | domain.tld |

## Post installation customization

* Prepare node for latest MU. Download the update repository to the Fuel Master node and use it as an update mirror. See details [here](https://docs.mirantis.com/openstack/fuel/fuel-7.0/maintenance-updates.html#maintenance-updates-for-mirantis-openstack-7-0)[[4]](#footnote-4).
* Install 8 plugins

|  |
| --- |
| # fuel plugins --install elasticsearch\_kibana-0.8-0.8.1-1.noarch.rpm  # fuel plugins --install influxdb\_grafana-0.8-0.8.1-1.noarch.rpm  # fuel plugins --install ldap-1.0-1.0.0-1.noarch.rpm  # fuel plugins --install lma\_collector-0.8-0.8.1-1.noarch.rpm  # fuel plugins --install lma\_infrastructure\_alerting-0.8-0.8.1-1.noarch.rpm  # fuel plugins --install local-ephemeral-1.0-1.0.0-1.noarch.rpm  # fuel plugins --install openldap-ad-1.0-1.0.1-1.noarch.rpm  # fuel plugins --install publicnet-vlan-1.0-1.0.0-1.noarch.rpm  # fuel plugins list  id | name | version | package\_version  ---|-----------------------------|---------|----------------  1 | elasticsearch\_kibana | 0.8.1 | 3.0.0  2 | influxdb\_grafana | 0.8.1 | 3.0.0  3 | lma\_collector | 0.8.1 | 3.0.0  4 | lma\_infrastructure\_alerting | 0.8.1 | 3.0.0  5 | ldap | 1.0.0 | 3.0.0  6 | openldap-ad | 1.0.1 | 3.0.0  7 | publicnet-vlan | 1.0.0 | 3.0.0  8 | local-ephemeral | 1.0.0 | 3.0.0 |

* Apply patch for ldap plugin

|  |
| --- |
| # vim 3.patch  --- /var/www/nailgun/plugins/ldap-1.0/deployment\_scripts/puppet/modules/plugin\_ldap/manifests/controller.pp.orig 2016-05-24 22:10:47.239070603 +0000  +++ /var/www/nailgun/plugins/ldap-1.0/deployment\_scripts/puppet/modules/plugin\_ldap/manifests/controller.pp 2016-05-24 17:41:13.145736977 +0000  @@ -122,6 +122,7 @@  "${domain}/ldap/group\_allow\_create": value => $group\_allow\_create;  "${domain}/ldap/group\_allow\_update": value => $group\_allow\_update;  "${domain}/ldap/group\_allow\_delete": value => $group\_allow\_delete;  + "${domain}/ldap/chase\_referrals": value => false;  } ~>  service { 'httpd':  name => "$apache::params::service\_name",  # patch -p0 -i 3.patch |

# OpenStack Environment Deployment

## Creation of environment:

Step by step instruction for installation of an OpenStack environment using Fuel is available [here](https://docs.mirantis.com/openstack/fuel/fuel-8.0/user-guide.html#start-create-env-ug)[[5]](#footnote-5) .

Following settings were used during installation of OpenStack:

|  |  |
| --- | --- |
| Name | [Customer]-Staging |
| OpenStack Release | Kilo on Ubuntu 14.04 |
| Compute | QEMU-KVM |
| Networking Setup | Neutron with tunneling segmentation |
| Storage Backends | Yes, use Ceph |
| Additional services | Sahara, Murano, Ceilometer (OpenStack Telemetry) |

# 

# Pre Deployment Customization

## Change notation for networks (via cli on Fuel-Master)

|  |
| --- |
| # fuel env  id | status | name | mode | release\_id | pending\_release\_id  ---|--------|------|------------|------------|-------------------  5 | new | t5 | ha\_compact | 2 | None  # fuel network --download --env 5  # sed -i 's@notation: cidr@notation: ip\_ranges@g' /root/network\_5.yaml  # fuel network --upload --env 5 |

# Adding and Removing Compute & Storage Nodes

In order to add or remove compute + cinder storage node to\from the environment you need to follow this guide [here](https://docs.mirantis.com/openstack/fuel/fuel-8.0/operations.html?highlight=add%20compute#adding-redeploying-and-replacing-nodes)[[6]](#footnote-6).

ATTENTION:

* Before removing compute node all VMs that running on target node need to be migrated to other compute nodes.

# Post Deployment Configuration

* Availability zones AZ1, AZ2 have been created both for nova and cinder. Note: ***cinder\_cross\_az\_attach=False*** has been set to prevent attach between instance and volume in different availability zones.

## [Extensions implemented in OpenStack to define the provider network]

The following provider extensions have been implemented at <...> to extend the Networking network resource with these attributes:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute name** | **Type** | **Default Value** | **Description** |
| provider:network\_type | String | N/A | The physical mechanism by which the virtual network is implemented. Possible values are flat, vlan, local, and gre, corresponding to flat networks, VLAN networks, local networks, and GRE networks as defined above. For <...>, this was set to VLAN. |
| provider:physical\_network | String | default | The name of the physical network over which the virtual network is implemented for flat and VLAN networks. For <...>, set to the Physical Network name provided. |
| provider:segmentation\_id | Integer | N/A | For VLAN networks, the VLAN VID on the physical network that realizes the virtual network. For <...>, this value was set to the VID’s provided by <...> for the initial thirteen networks. |

To view or set provider extended attributes, a client must be authorized for the extension:provider\_network:view and extension:provider\_network:set actions in the Networking policy configuration. The default Networking configuration authorizes both actions for users with the admin role. An authorized client or an administrative user can view and set the provider extended attributes through Networking API calls.

NOTE: Mirantis suggests creating **non shared** networks (1 network for 1 tenant). This suggestion will allow the individual tenant consumers to launch a VM in their tenant using either the OpenStack CLI or some kind of automation/orchestration tool such as RightScale and omit having to provide network information. The VM will be launched and connected automatically to the only network belonging to that tenant.

## 

## [Creating provider networks within OpenStack]

An administrative user with the proper privileges and credentials can perform the following steps after the <...> Networking team has created the correct VLAN and has configured the current 10g Network switch to recognize it in order to add the newly created VLAN to the <...> OpenStack cloud:

a. The administrator will need to know the ID of the tenants that the network is being provided to. When creating a subnet the administrator should use the ID of network created by the net-create command. An example of the network creation and subnet creation are provided below:

***root@node-1:~# neutron net-create --provider:physical\_network=physnet2 --provider:network\_type=vlan --provider:segmentation\_id=500 MDCTest***

***root@node-1:~# neutron subnet-create --tenant-id e88fa9ba254a45e6a9e186677b08388d MDCTest 172.16.0.0/24 --gateway 172.16.0.1***

NOTE: The tenant ID and network ID depicted above would be changed to reflect the desired tenant and ID of the network created.

b. Also you can use Horizon to create network. In order to do so, you will be needed to change projects each time you are creating a new network for the specific tenant.

You can create the networks and subnets from the Horizon Network Topology menu depicted below:

<...>

NOTE: You can not specify VLAN ID when creating network using Horizon.

# Cloud Verification Test Results

## Cli and Web UI verification results

|  |  |  |
| --- | --- | --- |
| **Check** | **Web UI** | **CLI** |
| Add new tenant and new user, login using created credentials | Done | Done |
| Upload an image | Done | Done |
| Add network to the tenant | Done | Done |
| Spawn new instance | Done | Done |
| Create new volume, attach to the instance | Done | Done |
| Login to the instance using VNC console and verify IP assignment and Internet/external network access from instance | Done |  |
| Check that volume is attached and writable | Done |  |

1. <https://docs.mirantis.com/openstack/fuel/fuel-8.0/user-guide.html#change-fuel-passwd-ug> [↑](#footnote-ref-1)
2. <https://docs.mirantis.com/openstack/fuel/fuel-8.0/pdf/Mirantis-OpenStack-8.0-OperationsGuide.pdf> [↑](#footnote-ref-2)
3. <https://docs.mirantis.com/openstack/fuel/fuel-8.0/fuel-install-guide.html#install-install-fuel-master-node> [↑](#footnote-ref-3)
4. <https://docs.mirantis.com/openstack/fuel/fuel-8.0/release-notes.html#maintenance-updates> [↑](#footnote-ref-4)
5. <https://docs.mirantis.com/openstack/fuel/fuel-8.0/user-guide.html#start-create-env-ug> [↑](#footnote-ref-5)
6. <https://docs.mirantis.com/openstack/fuel/fuel-8.0/operations.html?highlight=add%20compute#adding-redeploying-and-replacing-nodes> [↑](#footnote-ref-6)