Red Hat Reference Architecture Series

Running EAP 6 on RHEL OSP 7

Jacob Liberman

Version 1.0, 2015-08-05

Draft

Table of Contents

Draft

Draft

100 East Davie Street
Raleigh NC 27601 USA
Phone: +1 919 754 3700
Phone: 888 733 4281
Fax: +1 919 754 3701
PO Box 13588
Research Triangle Park NC 27709 USA

Linux is a registered trademark of Linus Torvalds. Red Hat, Red Hat Enterprise Linux and the Red Hat "Shadowman" logo are registered trademarks of Red Hat, Inc. in the United States and other countries. Microsoft and Windows are U.S. registered trademarks of Microsoft Corporation. UNIX is a registered trademark of The Open Group. Intel, the Intel logo and Xeon are registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries. All other trademarks referenced herein are the property of their respective owners.

© 2015 by Red Hat, Inc. This material may be distributed only subject to the terms and conditions set forth in the Open Publication License, V1.0 or later (the latest version is presently available at http://www.opencontent.org/openpub/).

The information contained herein is subject to change without notice. Red Hat, Inc. shall not be liable for technical or editorial errors or omissions contained herein.

Distribution of modified versions of this document is prohibited without the explicit permission of Red Hat Inc.

Distribution of this work or derivative of this work in any standard (paper) book form for commercial purposes is prohibited unless prior permission is obtained from Red Hat Inc.

The GPG fingerprint of the security@redhat.com key is: CA 20 86 86 2B D6 9D FC 65 F6 EC C4 21 91 80 CD DB 42 A6 0E

Send feedback to refarch-feedback@redhat.com

Comments and Feedback

In the spirit of open source, we invite anyone to provide feedback and comments on any reference architectures. Although we review our papers internally, sometimes issues or typographical errors are encountered. Feedback allows us to not only improve the quality of the papers we produce, but allows the reader to provide their thoughts on potential improvements and topic expansion to the papers. Feedback on the papers can be provided by emailing refarch-feedback@redhat.com. Please refer to the title within the email.

Staying In Touch

Join us on some of the popular social media sites where we keep our audience informed on new reference architectures as well as offer related information on things we find interesting.

Like us on Facebook:

https://www.facebook.com/rhrefarch

Follow us on Twitter:

https://twitter.com/RedHatRefArch

Plus us on Google+

https://plus.google.com/u/0/b/114152126783830728030/

Draft

Executive Summary

Red Hat Enterprise Linux OpenStack Platform (RHEL OSP) 7 is a reliable cloud infrastructure platform based on OpenStack. RHEL OSP 7 has a modular, service-based architecture designed for massive scalability. This document describes Red Hat's approach to scaling OpenStack. Core OpenStack services are managed and monitored in highly available clusters. A load balancer provides access to the service endpoints. There are no direct connections from the clients to the services.

Benefits of this approach include:

- Highly available state database and messaging queues
- Services can be scaled by adding controller nodes
- Service isolation simplifies troubleshooting
- Granular startup control satisfies service interdependencies

It also introduces OSP-d: a tool for deploying and managing OpenStack Platform. OSP-d combines features from various open source projects including OpenStack TripleO, RDO Manager, and Spinal Stack.

This reference architecture describes a realistic use case for deploying a multi-tier EAP application on a highly available OSP 7 cluster. The first section of this reference architecture introduces the principal components: OSP 7, OSP-d, and EAP. The second section describes the lab environment, hardware, and software versions used to test the reference architecture. The third section lists the steps performed by the Red Hat Systems Engineering team to deploy the reference architecture in their lab using the OSP director, production code, and bare metal servers.

Draft

Architecture Overview

Red Hat JBOSS Enterprise Application Platform

Red Hat® JBoss® Enterprise Application Platform (EAP) is a fully certified Java™ EE platform for developing and deploying enterprise applications.

Red Hat Enterprise Linux OpenStack Platform 7

Red Hat Enterprise Linux OpenStack Platform delivers an integrated foundation to create, deploy, and scale a secure and reliable public or private OpenStack cloud. Red Hat Enterprise Linux OpenStack Platform combines the world’s leading enterprise Linux and the fastest-growing cloud infrastructure into a single platform that can scale quickly without compromising on availability, security, or performance.

OpenStack Platform-Director

- TripleO
- Heat
- Puppet
- Hardware profile matching
- Network isolation
- Ceph storage customization

Server Roles

OSP-d can deploy the following server roles:

Control

provides the endpoint for REST-based API queries to the majority of the OpenStack services. These include Compute, Image, Identity, Block, Network, and Data processing. Although RHEL OSP allows for multiple, high availability cloud controllers, only one cloud controller is used in this reference architecture.

Compute

Compute node refers to an OpenStack server that runs a KVM hypervisor. It is responsible for running virtual machine instances. In this reference architecture, Hadoop clusters are instantiated across multiple compute nodes. By default a new instance is spawned on the compute node with the most free memory in a round robin fashion.

Draft

Swift Storage

Understanding the Swift ring is central to understanding the role of a server in the Swift cluster. A Swift ring represents a mapping between the names of entities stored on disk and their physical location. There are separate rings for accounts, containers, and objects. When a component needs to interact with an object, container, or account, it interacts with the appropriate ring to determine the target's location in the cluster.

Block Storage

definition

Ceph Storage

definition

Network Isolation

A typical OpenStack deployment includes several network roles. In some cases the roles overlap across the same physical interfaces or switches. In others each role has a dedicated network interface and switch. This reference architecture uses the following networks:

1. External network – the external network is used to perform system maintenance tasks such as installing software. In a private cloud scenario, users access the cloud infrastructure via the external network.
2. Service network – this network exposes the OpenStack APIs. It also handles inter-service communication between the OpenStack services and schedulers.
3. Tenant network – virtual machines communicate over this network within the cloud deployment. The addressing requirements of this network depend on the plugin that is used.
4. Storage Network – this network is dedicated for storage traffic between the Swift servers and the OpenStack servers.

High Availability

Conceptual Diagram of the Solution Stack

Error: Reference source not found Error: Reference source not found depicts the solution stack including networks, server roles, and service placement. Section Error: Reference source not found Error: Reference source not found shares complete details.

Draft

Reference Architecture Configuration Details

Use case

Objective

Work flow

Environment

The reference architecture environment consists of the components required to build a small Red Hat Enterprise Linux OpenStack Platform cloud infrastructure. It includes small form factor servers for the OpenStack servers and Swift storage servers with more internal storage capacity.

Server Roles

Network Topology

Figure 2 shows the network topology of this reference architecture.

All eight servers communicate via the lab network switch on the management network. The management network uses IP addresses in the 10.19.137.0/24 range.

The tenant network carries communication between virtual machines and software-defined networking components. It is the private network over which the instances communicate. In this reference architecture, a network switch connected to 10 GB interfaces on the compute nodes is tagged to VLAN IDs 1000:1010 for tenant communication.

> The tenant network carries tenant network traffic over tagged VLANs. The interfaces connected to this network are not configured with IPv4 addresses by the OpenStack administrator. Instead, instances and services are allocated addresses within user-created subnets on tenant networks. Network namespaces prevent different users' subnets from conflicting with each other or with the infrastructure's own subnets.

All Swift storage communication occurs via a second 10Gb storage network switch on the 172.31.0.0/16 network. This network delivers the Object storage service communication and delivery. The Service network carries service requests to the service listeners. These include the various schedulers and agents deployed in the OpenStack environment. The service traffic is segmented from the tenant and management traffic. The service network interfaces are assigned IP addresses in the 172.16.2.0/24 range.

> This reference architecture uses four physical networks. However it is possible to deploy supported OpenStack solutions with more or fewer networks.

Draft

Network Configuration

This reference architecture uses the network isolation feature to segment openstack communication by type. The following network traffic types are segmented:

- Provisioning
- Internal API
- Storage
- Storage Management
- Tenant
- External

In some cases more than one traffic type will share a physical interface. This is because there are more traffic types than physical interfaces in the servers. The network isolation feature uses both tagged and native VLANs to segment network both within and across interfaces.

Each server has two Gigabit interfaces (nic1:2) and two 10-Gigabit interfaces (nic3:4).

Table 1. Network isolation

Role	Interface	Network	VLAN ID	VLAN Type
Undercloud	nic1	4041	External	Native
	nic2	168	Provisioning	Native
	nic3	4042	Storage Mgmt	Tagged
	nic3	4043	Tenant	Native
	nic4	4044	Internal API	Tagged
	nic4	4045	Storage	Native
Control	nic1	4041	External	Native
	nic2	168	Provisioning	Native
	nic3	4042	Storage Mgmt	Tagged
	nic3	4043	Tenant	Native
	nic4	4044	Internal API	Tagged
	nic4	4045	Storage	Native

Role	Interface	Network	VLAN ID	VLAN Type
Compute	nic1	4041	External	Native
	nic2	168	Provisioning	Native
	nic3	4042	Storage Mgmt	Tagged
	nic3	4043	Tenant	Native
	nic4	4044	Internal API	Tagged
	nic4	4045	Storage	Native
Ceph storage	nic1	4041	External	Native
	nic2	168	Provisioning	Native
	nic3	4042	Storage Mgmt	Tagged
	nic3	4043	Tenant	Native
	nic4	4044	Internal API	Tagged
	nic4	4045	Storage	Native

Install OSP 7 with OSP-D

This section lists the steps that were followed to install and configure OSP 7 with OSP-d in the Red Hat Systems Engineering lab.

Prepare the undercloud server

1. Install the operating system
2. Activate required channels
3. Install the osc plugin.

```
# rpm -q python-rdomanager-oscplugin
python-rdomanager-oscplugin-0.0.8-44.el7ost.noarch
```

1. Set the hostname

```
+ hostnamectl set-hostname rhos0.osplocal
+ hostnamectl set-hostname --transient rhos0.osplocal
+ export HOSTNAME=rhos0.osplocal
+ HOSTNAME=rhos0.osplocal
+ hostname
rhos0.osplocal
```

1. Create the stack user

```
+ useradd stack
+ echo stack:password
+ chpasswd
+ echo 'stack ALL=(root) NOPASSWD:ALL'
+ tee -a /etc/sudoers.d/stack
stack ALL=(root) NOPASSWD:ALL
+ chmod 0440 /etc/sudoers.d/stack
+ id stack
uid=1000(stack) gid=1000(stack) groups=1000(stack)
```

Deploy the Undercloud

1. Switch to the stack user account.

```
# su - stack
```

1. The file undercloud.conf contains configuration data for the undercloud installation. Create the undercloud.conf.

```
[DEFAULT]

image_path = .
local_ip = 192.0.2.1/24
#undercloud_public_vip = 192.0.2.2
#undercloud_admin_vip = 192.0.2.3
#undercloud_service_certificate =
local_interface = eno4
masquerade_network = 192.0.2.0/24
dhcp_start = 192.0.2.5
dhcp_end = 192.0.2.24
network_cidr = 192.0.2.0/24
network_gateway = 192.0.2.1
discovery_interface = br-ctlplane
discovery_iprange = 192.0.2.100,192.0.2.120
discovery_runbench = false
undercloud_debug = true

[auth]

undercloud_db_password =
undercloud_admin_token =
undercloud_admin_password =
undercloud_glance_password =
undercloud_heat_encryption_key =
undercloud_heat_password =
undercloud_neutron_password =
undercloud_nova_password =
undercloud_ironic_password =
undercloud_tuskar_password =
undercloud_ceilometer_password =
undercloud_ceilometer_metering_secret =
undercloud_ceilometer_snmpd_user =
undercloud_ceilometer_snmpd_password =
undercloud_swift_password =
undercloud_rabbit_cookie =
undercloud_rabbit_password =
undercloud_rabbit_username =
undercloud_heat_stack_domain_admin_password =
undercloud_swift_hash_suffix =
```

1. Install the undercloud.

```
$ openstack undercloud install 2> uc.err
```

1. Source stackrc to set environment variables.

```
$ source stackrc

$ env | grep OS_
OS_PASSWORD=8597924e5cd20db4b34a547ca6975c4c3fe4f691
OS_AUTH_URL=http://192.0.2.1:5000/v2.0
OS_USERNAME=admin
OS_TENANT_NAME=admin
OS_NO_CACHE=True
```

1. Check status of openstack-services. All services should be active.

```
$ openstack-service status
neutron-dhcp-agent (pid 16624) is active
neutron-openvswitch-agent (pid 17874) is active
neutron-server (pid 16672) is active
openstack-ceilometer-alarm-evaluator (pid 16288) is active
openstack-ceilometer-alarm-notifier (pid 16228) is active
openstack-ceilometer-api (pid 16257) is active
openstack-ceilometer-central (pid 16197) is active
openstack-ceilometer-collector (pid 16168) is active
openstack-ceilometer-notification (pid 16139) is active
openstack-glance-api (pid 17139) is active
openstack-glance-registry (pid 17074) is active
openstack-heat-api-cfn (pid 17903) is active
openstack-heat-api-cloudwatch (pid 18158) is active
openstack-heat-api (pid 18062) is active
openstack-heat-engine (pid 18025) is active
openstack-ironic-api (pid 14801) is active
openstack-ironic-conductor (pid 19167) is active
openstack-ironic-discoverd-dnsmasq (pid 20046) is active
openstack-ironic-discoverd (pid 20052) is active
openstack-keystone (pid 16786) is active
openstack-nova-api (pid 17272) is active
openstack-nova-compute (pid 19939) is active
openstack-nova-conductor (pid 17459) is active
openstack-nova-consoleauth (pid 17236) is active
openstack-nova-scheduler (pid 17421) is active
openstack-swift-account-auditor (pid 15664) is active
openstack-swift-account-reaper (pid 15639) is active
openstack-swift-account-replicator (pid 15954) is active
openstack-swift-account (pid 15979) is active
openstack-swift-container-auditor (pid 15751) is active
openstack-swift-container-replicator (pid 16043) is active
openstack-swift-container-updater (pid 16593) is active
openstack-swift-container (pid 16068) is active
openstack-swift-object-auditor (pid 15834) is active
openstack-swift-object-replicator (pid 16432) is active
openstack-swift-object-updater (pid 15859) is active
openstack-swift-object (pid 16403) is active
openstack-swift-proxy (pid 16334) is active
openstack-tuskar-api (pid 20076) is active
```

1. Get the images
2. Upload the images

```
$ openstack overcloud image upload

$ openstack image list
+--------------------------------------+------------------------+
| ID                                   | Name                   |
+--------------------------------------+------------------------+
| 10287a14-ed9d-4767-9088-1b8132f6b4bb | bm-deploy-ramdisk      |
| 2f237f37-7203-492e-b99f-3f64b6baa7e4 | bm-deploy-kernel       |
| 3984add9-d298-4598-904f-99d82531121a | overcloud-full         |
| be617ff4-853d-4b03-83ea-d4e10bc8399f | overcloud-full-initrd  |
| bdbd5a52-2307-4baa-abc7-12370b8ed396 | overcloud-full-vmlinuz |
+--------------------------------------+------------------------+
```

Deploy the overcloud

This section describes steps for deploying the overcloud.

1. Create the host definition file. The openstack-ironic-discoverd service uses this file to discover nodes and populate the ironic database.

   > The example below is truncated for brevity. Appendix [link to appendix] contains the full file.

Draft

```
{
  "nodes": [
    {
      "pm_password": "100Mgmt-",
      "pm_type": "pxe_ipmitool",
      "mac": [
        "d4:ae:52:b2:20:d2"
      ],
      "cpu": "24",
      "memory": "49152",
      "disk": "500",
      "arch": "x86_64",
      "pm_user": "root",
      "pm_addr": "10.19.143.153"
    },
    {
      "pm_password": "100Mgmt-",
      "pm_type": "pxe_ipmitool",
      "mac": [
        "54:9F:35:F6:70:70"
      ],
      "cpu": "32",
      "memory": "98304",
      "disk": "130",
      "arch": "x86_64",
      "pm_user": "root",
      "pm_addr": "10.19.143.37"
     }
  ]
}
```

1. Register the nodes with ironic.

```
$ *openstack baremetal import --json ~/instackenv.json*

$ *openstack baremetal list*
+--------------------------------------+------+---------------+-------------
+-----------------+-------------+
| UUID                                 | Name | Instance UUID | Power
State | Provision State | Maintenance |
+--------------------------------------+------+---------------+-------------
+-----------------+-------------+
| 1e8d1fbc-7355-4de3-a1cb-5de7c0f54808 | None | None          | power
off   | available       | False       |
| de5d489a-8232-4a98-b43b-531f6da21866 | None | None          | power
off   | available       | False       |
| c9eb5e27-5815-4f9e-95f9-7854753951d9 | None | None          | power
off   | available       | False       |
| 4eedcd08-b901-433a-99c1-239da5279b9a | None | None          | power
off   | available       | False       |
| 956418e8-5c64-45cd-af66-8e771077b987 | None | None          | power
off   | available       | False       |
| f89038eb-0631-4682-b635-10ac1cf3a265 | None | None          | power
off   | available       | False       |
| 921fbec6-7fa0-4db4-b3a4-38878aa0694f | None | None          | power
off   | available       | False       |
| 1cb6ac7e-8cf2-4aff-80e9-f8e103d30fc7 | None | None          | power
off   | available       | False       |
| ba083e12-2519-4eed-9489-3f2abcf7832f | None | None          | power
off   | available       | False       |
| bea915fd-c4bc-4931-b8b6-ad175be42f4f | None | None          | power
off   | available       | False       |
| a609e230-7cdd-4ee5-a660-f1e55cf4f5a7 | None | None          | power
off   | available       | False       |
+--------------------------------------+------+---------------+-------------
+-----------------+-------------+
```

1. Assign a kernel and ramdisk to the nodes

```
$ *openstack baremetal configure boot*
```

1. Introspect the nodes to discover their hardware attributes.

```
$ *openstack baremetal introspection bulk start*
```

Bulk introspection time will vary based on the number of physical nodes and their individual boot times. For this reference architecture bulk introspection lasted approximately 3 minutes per node.

1. Use journalctl to view introspection progress in a separate terminal.

```
$ *sudo journalctl -l -u openstack-ironic-discoverd -u
openstack-ironic-discoverd-dnsmasq -u openstack-ironic-conductor |
tailf*
Aug 06 10:23:00 rhos0.osplocal ironic-conductor[19167]: 2015-08-06
10:23:00.307 19167 DEBUG oslo_concurrency.processutils [-] Running cmd
(subprocess): ipmitool -I lanplus -H 10.19.143.37 -L ADMINISTRATOR -U
root -R 12 -N 5 -f /tmp/tmpm55PAU power status execute
/usr/lib/python2.7/site-packages/oslo_concurrency/processutils.py:199
Aug 06 10:23:00 rhos0.osplocal dnsmasq-dhcp[20046]:
DHCPREQUEST(br-ctlplane) 192.0.2.105 54:9f:35:f6:70:3c
Aug 06 10:23:00 rhos0.osplocal dnsmasq-dhcp[20046]:
DHCPACK(br-ctlplane) 192.0.2.105 54:9f:35:f6:70:3c
Aug 06 10:23:00 rhos0.osplocal ironic-conductor[19167]: 2015-08-06
10:23:00.366 19167 DEBUG oslo_concurrency.processutils [-] CMD
"ipmitool -I lanplus -H 10.19.143.37 -L ADMINISTRATOR -U root -R 12 -N
5 -f /tmp/tmpm55PAU power status" returned: 0 in 0.059s execute
/usr/lib/python2.7/site-packages/oslo_concurrency/processutils.py:225
```

1. Verify nodes completed introspection without errors.

```
$ openstack baremetal introspection bulk status
+--------------------------------------+----------+-------+
| Node UUID                            | Finished | Error |
+--------------------------------------+----------+-------+
| 1e8d1fbc-7355-4de3-a1cb-5de7c0f54808 | True     | None  |
| de5d489a-8232-4a98-b43b-531f6da21866 | True     | None  |
| c9eb5e27-5815-4f9e-95f9-7854753951d9 | True     | None  |
| 4eedcd08-b901-433a-99c1-239da5279b9a | True     | None  |
| 956418e8-5c64-45cd-af66-8e771077b987 | True     | None  |
| f89038eb-0631-4682-b635-10ac1cf3a265 | True     | None  |
| 921fbec6-7fa0-4db4-b3a4-38878aa0694f | True     | None  |
| 1cb6ac7e-8cf2-4aff-80e9-f8e103d30fc7 | True     | None  |
| ba083e12-2519-4eed-9489-3f2abcf7832f | True     | None  |
| bea915fd-c4bc-4931-b8b6-ad175be42f4f | True     | None  |
| a609e230-7cdd-4ee5-a660-f1e55cf4f5a7 | True     | None  |
+--------------------------------------+----------+-------+
```

Configure hardware profiles

1. Create the default flavor for baremetal deployments.

```
$ *openstack flavor create --id auto --ram 4096 --disk
40 --vcpus 1 baremetal*
+----------------------------+--------------------------------------+
| Field                      | Value                                |
+----------------------------+--------------------------------------+
| OS-FLV-DISABLED:disabled   | False                                |
| OS-FLV-EXT-DATA:ephemeral  | 0                                    |
| disk                       | 40                                   |
| id                         | 61209bd2-d573-4e9c-b0ec-8be332c4582f |
| name                       | baremetal                            |
| os-flavor-access:is_public | True                                 |
| ram                        | 4096                                 |
| rxtx_factor                | 1.0                                  |
| swap                       |                                      |
| vcpus                      | 1                                    |
+----------------------------+--------------------------------------+
```

1. Set properties for the baremetal flavor.

```
$ *openstack flavor set --property "cpu_arch"="x86_64"
--property "capabilities:boot_option"="local" baremetal*
+----------------------------+-----------------------------------------------------+
| Field                      | Value
+----------------------------+-----------------------------------------------------+
| OS-FLV-DISABLED:disabled   | False
| OS-FLV-EXT-DATA:ephemeral  | 0
| disk                       | 40
| id                         | 61209bd2-d573-4e9c-b0ec-8be332c4582f
| name                       | baremetal
| os-flavor-access:is_public | True
| properties                 | capabilities:boot_option='local', cpu_arch='x86_64'
| ram                        | 4096
| rxtx_factor                | 1.0
| swap                       |
| vcpus                      | 1
+----------------------------+-----------------------------------------------------+
```

.

Configure network isolation

Configure Ceph Storage

Deploy the overcloud servers

Configure controller fencing

Test the overcloud

Draft

Install and Configure EAP 6

This ection describes the steps to install and configure an example EAP application on the deployed cloud. The example EAP application is a multi-tier web application with a shopping cart.

Define heat templates

Deploy instances

Install EAP

Test EAP server

Draft

Conclusion

Draft

Appendix A: Revision History

Draft

Appendix B: References

Draft

Appendix C: Security reference

Some text

Firewall Configuration ~~~~~~~~ Allowed firewall ports by role.

Table 2. Allowed Ports by Role

Role	Port	Service
Undercloud	xx	xyz
Control	xx	xyz
Compute	xx	xyz
Ceph storage	xx	xyz

SELinux Configuration

RHEL OSP-d 7 supports SELinux in enforcing and permissive modes.

Table 3. Supported SELinux Package Versions

Package	Version
libselinux	2.2.2-6
selinux-policy	3.13.1-23
selinux-policy-targeted	3.13.1-23
openstack-selinux	0.6.27-1

Appendix D: Hardware specifications

Table 4. Hardware specifications

Count	Model	Description
8	Dell PowerEdge M520	2x Intel Xeon CPU E5-2450 0 @ 2.10GHz, Broadcom 5720 1Gb Dual Port LOMs, Broadcom 57810S-k Dual Port 10Gb NIC, 6x DDR3 8192 MB @1333 MHZ DIMMs, 2 x 146GB SAS internal disk drives
4	Dell PowerEdge R520	2x Intel® Xeon® CPU X5650 @ 2.67 GHz (6 core), 2 x Broadcom NetXtreme II BCM5709S Gb Ethernet, 2x Emulex Corporation OneConnect 10Gb NIC, 6 x DDR3 8192 MB @1333 MHZ DIMMs, 12x 146GB SAS internal disk drives
1	Dell PowerEdge M720	2x Intel® Xeon® CPU X5650 @ 2.67 GHz (6 core), 2 x Broadcom NetXtreme II BCM5709S Gb Ethernet, 2x Emulex Corporation OneConnect 10Gb NIC, 6 x DDR3 8192 MB @1333 MHZ DIMMs, 12x 146GB SAS internal disk drives

Appendix E: Service placement

This table lists the service placement for all OpenStack services.

Table 5. Service placement

Role	Count	Services
Undercloud	1	rabbitmq
		neutron-server
		openstack-cinder-api
		openstack-cinder-scheduler
		openstack-cinder-volume
		openstack-glance-api
		openstack-glance-registry
		openstack-heat-api-cfn
		openstack-heat-api
		openstack-heat-engine
		openstack-keystone
		openstack-nova-api
		openstack-nova-cert
		openstack-nova-conductor
		openstack-nova-consoleauth
		openstack-nova-novncproxy
		openstack-nova-scheduler
Control	3	neutron-dhcp-agent
		neutron-l3-agent
		neutron-metadata-agent
		neutron-openvswitch-agent
Compute	4	neutron-openvswitch-agent
		openstack-nova-compute

Role	Count	Services
Ceph Storage	3	openstack-swift-account
		openstack-swift-container
		openstack-swift-object

Draft

Appendix F: Required channels

Red Hat Enterprise Linux OpenStack Platform is available via Red Hat Network Certificate Server repositories.

Table 6. Required channels

Channel	Source
rhel-x86_64-server-7	RHN Classic
rhel-x86_64-server-7-ost-6	RHN Classic
rhel-7-server-rpms	RHN Certificate
rhel-7-server-openstack-6.0-rpms	RHN Certificate
rhel-7-server-rh-common-rpms	RHN Certificate

This reference architecture uses a local satellite server for deployments and updates.

Draft

Appendix G: instackenv.json


```
{
  "nodes": [
    {
      "pm_password": "100Mgmt-",
      "pm_type": "pxe_ipmitool",
      "mac": [
        "d4:ae:52:b2:20:d2"
      ],
      "cpu": "24",
      "memory": "49152",
      "disk": "500",
      "arch": "x86_64",
      "pm_user": "root",
      "pm_addr": "10.19.143.153"
    },
    {
      "pm_password": "100Mgmt-",
      "pm_type": "pxe_ipmitool",
      "mac": [
        "d4:ae:52:b2:28:95"
      ],
      "cpu": "24",
      "memory": "49152",
      "disk": "500",
      "arch": "x86_64",
      "pm_user": "root",
      "pm_addr": "10.19.143.154"
    },
    {
      "pm_password": "100Mgmt-",
      "pm_type": "pxe_ipmitool",
      "mac": [
        "d4:ae:52:b2:1c:37"
      ],
      "cpu": "24",
      "memory": "49152",
      "disk": "500",
      "arch": "x86_64",
      "pm_user": "root",
      "pm_addr": "10.19.143.151"
    },
    {
      "pm_password": "100Mgmt-",
      "pm_type": "pxe_ipmitool",
```

```
        "mac": [
          "d4:ae:52:b2:2e:80"
        ],
        "cpu": "24",
        "memory": "49152",
        "disk": "500",
        "arch": "x86_64",
        "pm_user": "root",
        "pm_addr": "10.19.143.152"
      },
      {
        "pm_password": "100Mgmt-",
        "pm_type": "pxe_ipmitool",
        "mac": [
          "54:9f:35:f6:70:22"
        ],
        "cpu": "32",
        "memory": "98304",
        "disk": "130",
        "arch": "x86_64",
        "pm_user": "root",
        "pm_addr": "10.19.143.31"
      },
      {
        "pm_password": "100Mgmt-",
        "pm_type": "pxe_ipmitool",
        "mac": [
          "54:9F:35:F6:70:2F"
        ],
        "cpu": "32",
        "memory": "98304",
        "disk": "130",
        "arch": "x86_64",
        "pm_user": "root",
        "pm_addr": "10.19.143.32"
      },
      {
        "pm_password": "100Mgmt-",
        "pm_type": "pxe_ipmitool",
        "mac": [
          "54:9F:35:F6:70:3C"
        ],
        "cpu": "32",
        "memory": "98304",
        "disk": "130",
        "arch": "x86_64",
        "pm_user": "root",
        "pm_addr": "10.19.143.33"
```

```
    },
    {
      "pm_password": "100Mgmt-",
      "pm_type": "pxe_ipmitool",
      "mac": [
        "54:9F:35:F6:70:49"
      ],
      "cpu": "32",
      "memory": "98304",
      "disk": "130",
      "arch": "x86_64",
      "pm_user": "root",
      "pm_addr": "10.19.143.34"
    },
    {
      "pm_password": "100Mgmt-",
      "pm_type": "pxe_ipmitool",
      "mac": [
        "54:9F:35:F6:70:56"
      ],
      "cpu": "32",
      "memory": "98304",
      "disk": "130",
      "arch": "x86_64",
      "pm_user": "root",
      "pm_addr": "10.19.143.35"
   },
    {
      "pm_password": "100Mgmt-",
      "pm_type": "pxe_ipmitool",
      "mac": [
        "54:9F:35:F6:70:63"
      ],
      "cpu": "32",
      "memory": "98304",
      "disk": "130",
      "arch": "x86_64",
      "pm_user": "root",
      "pm_addr": "10.19.143.36"
     },
    {
      "pm_password": "100Mgmt-",
      "pm_type": "pxe_ipmitool",
      "mac": [
        "54:9F:35:F6:70:70"
      ],
      "cpu": "32",
      "memory": "98304",
```

```
      "disk": "130",
      "arch": "x86_64",
      "pm_user": "root",
      "pm_addr": "10.19.143.37"
    }
  ]
}
```


Draft

Appendix H: Software versions

Table 7. Software Versions

Host	Software	Version
Control	openstack-cinder	2014.2.2-2
	openstack-dashboard	2014.2.2-2
	openstack-dashboard-theme	2014.2.2-2
	openstack-glance	2014.2.2-1
	openstack-heat-api	2014.2.2-1
	openstack-heat-common	2014.2.2-1
	openstack-heat-engine	2014.2.2-1
	openstack-keystone	2014.2.2-1
	openstack-neutron	2014.2.2-5
	openstack-nova-api	2014.2.2-5
	openstack-nova-cert	2014.2.2-5
	openstack-nova-common	2014.2.2-19
	openstack-nova-conductor	2014.2.2-19
	openstack-nova-console	2014.2.2-19
	openstack-nova-novncproxy	2014.2.2-19
	openstack-nova-scheduler	2014.2.2-19
	openstack-puppet-modules	2014.2.13-2
	openstack-sahara	2014.2.2-1
	openstack-sahara-doc	2014.2.2-1
	openstack-selinux	0.6.27-1
	openstack-swift	2.2.0-3
	openstack-swift-plugin-swift3	1.7-3
	openstack-swift-proxy	2.2.0-3
	openstack-utils	2014.2-1
	python-django-openstack-auth	1.1.7-4
	python-openstackclient	1.0.1-1

Host	Software	Version
Compute	openstack-neutron	2014.2.2-5
	openstack-neutron-openvswitch	2014.2.2-5
	openstack-nova-common	2014.2.2-19
	openstack-nova-compute	2014.2.2-19
	openstack-selinux	0.6.27-1
	openstack-utils	2014.2-1
	python-neutron	2014.2.2-5
	python-neutronclient	2.3.9-1
	openstack-neutron	2014.2.2-5
	openstack-neutron-ml2	2014.2.2-5
	openstack-neutron-openvswitch	2014.2.2-5
	openstack-selinux	0.6.27-1
	openstack-utils	2014.2-1
	python-neutron	2014.2.2-5
	python-neutronclient	2.3.9-1
Ceph Storage	openstack-selinux	0.6.27-1
	openstack-swift	2.2.0-3
	openstack-swift-account	2.2.0-3
	openstack-swift-container	2.2.0-3
	openstack-swift-object	2.2.0-3
	openstack-utils	2014.2-1