Red Hat Reference Architecture Series

Deploying an OpenShift Enterprise 3 Distributed Architecture

Scott Collier

Version 1.0, 2015-09-09

DRAFT

Table of Contents

DRAFT

100 East Davie Street
Raleigh NC 27601 USA
Phone: +1 919 754 3700
Phone: 888 733 4281
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.

Ceph is a registered trademark of Red Hat, Inc.

UNIX is a registered trademark of The Open Group.

Docker and the Docker logo are trademarks or registered trademarks of Docker, Inc.

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 architecture. 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

1. Executive Summary

Red Hat OpenShift 3 is built around a core of application containers powered by Docker, with orchestration and management provided by Kubernetes, on a foundation of Atomic host and Enterprise Linux. OpenShift Origin is the upstream community project that brings it all together along with extensions, to accelerate application development and deployment.

This reference environment provides a comprehensive example demonstrating how OpenShift 3 can be set up to take advantage of the Red Hat High Availability Add-On to provide an active / passive configuration. The configuration consists of three OpenShift masters and two OpenShift nodes in a single datacenter environment. In addition to the configuration, operational management tasks are shown to demonstrate functionality.

DRAFT

2. Components

This section provides a high-level representation of the components within this reference architecture. The focus is on components used for deploying a Red Hat OpenShift 3 Environment using the Red Hat High Availability Add-On running on Red Hat Enterprise Linux 7 x86_64.

2.1. *OpenShift*

Masters:

OpenShift Master - The master validates and configures the data for pods, services, and replication controllers. It also assigns pods to nodes and synchronizes pod information with service configuration.

etcd - The reliable storage backend for OpenShift to maintain cluster state. This etcd deployment is configured as a 2n+1 cluster for a recommended cluster size providing fault tolerance.

Pacemaker - The core of the Red Hat Enterprise Linux highly available technology, provides consensus, fencing and service management. It runs on all master nodes and ensures that all active-passive components have one instance running.

Virtual IP - a single point of contact, which is not a single point of failure, for all OpenShift clients that:

- Cannot be configured with all master service endpoints.
- Does not know how to load balance across multiple masters nor retry failed master service connections.

Nodes:

OpenShift Node - A node provides the runtime environments for containers. Each node in a OpenShift cluster has the required services to be managed by the master.

Docker - The open source application engine that is the core unit of packaging in OpenShift.

DRAFT

Components - HA Configuration shows the components of OpenShift and how they are clustered - either active / active or active / passive.

Table 1. Components - HA Configuration

Role	Style	Notes
etcd	Active-Active	Fully redundant deployment with load balancing
openshift-master	Active-Passive	One active at a time, managed by Pacemaker
Pacemaker	Active-Active	Fully redundant deployment
Virtual IP	Active-Passive	One active at a time, managed by Pacemaker

2.2. *Bind*

Berkeley Internet Name Daemon (BIND) is a implementation of the Domain Name System (DNS) protocols. BIND enables a human or computer to look-up another computer on the basis of a name. BIND contains three parts:

- DNS Server – answers queries that are sent to it.
- DNS Resolver Library – library that can be added to other programs to provide the ability to resolve names ensuring DNS standards are being followed.
- Tools for Testing Servers – Tools to test server environments, such as dig, nslookup and host.

BIND is an integral part of a successful OpenShift Enterprise deployment/environment. OpenShift Enterprise requires a properly configured wildcard DNS zone that resolves to the IP address of the OpenShift router. For more information, please refer to the installation guide: https://access.redhat.com/documentation/en/openshift-enterprise/version-3.0/openshift-enterprise-30-administrator-guide/chapter-1-installing

2.3. *High Availability Add-on*

Red Hat high availability clusters provide highly available services by eliminating single points of failure and failing over services from one cluster node to another in case a node becomes inoperative. Typically, services in a high availability cluster read and write data (via read-write mounted file systems). Therefore, a high availability cluster must maintain data integrity as one cluster node takes over control of a service from another cluster node. Node failures in a high availability cluster are not visible from clients outside the cluster (high availability clusters are sometimes referred to as failover clusters). The High Availability Add-On provides high availability clustering through its High Availability Service Management component, Pacemaker.

In this configuration, the highly available cluster does not leverage a shared filesystem. Instead, this cluster only utilizes virtual IP functionality of the Red Hat High Availability suite. For more information on the Red Hat High Availability add-on, refer to the product documentation: https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/High_Availability_Add-On_Administration/index.html

DRAFT

3. Reference Architecture Configuration

This chapter describes the reference architecture environment that is deployed which enables the configuration of a highly available OpenShift Enterprise 3 environment.

3.1. *Environment*

This environment consists of a single Dell M1000e chassis with five Dell M520 blades configured as described in the Hardware Details table. There are three OpenShift masters and two OpenShift nodes. The hardware is identical between all the servers.

3.2. *Hardware Details*

Within this reference environment, all the servers are located within the same blade chassis. No external storage is required for the deployment, however, consult with your administrators to properly scope out the requirements for a highly available OpenShift PaaS.

Table 2. Hardware Details

Dell Blades	Specifications
Dell M520 x5	4 x Broadcom Corporation NetXtreme BCM5720
	2 Socket, 8 Core (16 Threads w/HyperThreading) Intel® Xeon® CPU E5-2450 0 @ 2.10GHz
	96 GB of memory, DDR3 16GB @ 1600 MHz
	2 x 136 GB SAS internal disk drives

3.3. *Software Version Details*

The following tables provide the installed software versions for the different servers that make up the Red Hat OpenShift highly available reference environment.

Table 3. RHEL OSEv3 HA Details

Software	Version
Red Hat Enterprise Linux 7.1 x86_64	kernel 3.10.0-229.el7.x86_64
pacemaker	1.1.12-22.el7_1.2.x86_64
openshift	3.0.1.0-1.git.525.eddc479.el7ose.x86_64
docker	1.7.1-108.el7.x86_64
ansible	1.9.2-1.el7.noarch
openshift ansible installer	release: branch v3.0.1-1

3.4. *Required Channels*

The following channels must be subscribed to in order to access the content needed to deploy this configuration.

Table 4. Required Channels - OSEv3 Master Servers

Channel	Repository Name
Red Hat Enterprise Linux 7 Server (RPMs)	rhel-7-server-rpms
Red Hat Enterprise Linux High Availability (for RHEL 7 Server) (RPMs)	rhel-ha-for-rhel-7-server-rpms
Red Hat OpenShift Enterprise 3.0 (RPMs)	rhel-7-server-ose-3.0-rpms
Red Hat Enterprise Linux 7 Server - Extras (RPMs)	rhel-7-server-extras-rpms

Table 5. Required Channels - OSEv3 Nodes

Channel	Repository Name
Red Hat Enterprise Linux 7 Server (RPMs)	rhel-7-server-rpms
Red Hat OpenShift Enterprise 3.0 (RPMs)	rhel-7-server-ose-3.0-rpms
Red Hat Enterprise Linux 7 Server - Extras (RPMs)	rhel-7-server-extras-rpms

Table 6. Host Connectivity

Hostname	IP Address
master-1.cloud.example.com	10.x.x.130
master-2.cloud.example.com	10.x.x.131
master-3.cloud.example.com	10.x.x.132
node-1.cloud.example.com	10.x.x.133
node-1.cloud.example.com	10.x.x.134
vip.cloud.example.com	10.x.x.135

Table 7. iDRAC Connectivity

Hostname	iDRAC IP Address
master-1.cloud.example.com	10.x.x.82
master-2.cloud.example.com	10.x.x.83
master-3.cloud.example.com	10.x.x.84

DRAFT

4. Deploying the Infrastructure

The following section describes in detail the installation procedures required for the reference environment.

4.1. Prerequisites

The following prerequisites are not required, but highly recommended, prior to deployment of this environment.

The recommended prerequisites consist of:

- Red Hat Satellite 6 - is a life-cycle management platform for your Red Hat Enterprise Linux infrastructure. It provides the tools to efficiently deploy, update, monitor, and manage systems.
- DNS - specifically, a DNS zone that is setup to support the wildcard DNS zone that OpenShift requires for proper application functionality.

This reference environment takes advantage of Red Hat Satellite 6 and DNS zoning as part of the infrastructure deployment process.

4.2. Name Resolution

Proper name resolution is configured for the servers and applications. The requirements for the servers are to have forward and reverse lookups capabilties. Ensure that hostnames can be resolved by issuing the following commands.

```
# for i in seq 1 3; do host ose-master-$i; done
ose-master-1.cloud.example.com has address 10.x.x.5
ose-master-2.cloud.example.com has address 10.x.x.6
ose-master-3.cloud.example.com has address 10.x.x.7

# for i in seq 5 7; do host 10.x.x.$i; done
5.x.x.10.in-addr.arpa domain name pointer ose-master-1.cloud.example.com.
6.x.x.10.in-addr.arpa domain name pointer ose-master-2.cloud.example.com.
7.x.x.10.in-addr.arpa domain name pointer ose-master-3.cloud.example.com.
```

OpenShift applications require proper wildcard DNS configuration. The installation section of the OpenShift Administrator guide provides more information about this requirement. https://access.redhat.com/documentation/en/openshift-enterprise/version-3.0/openshift-enterprise-30-administrator-guide/chapter-1-installing

4.3. Install Red Hat Enterprise Linux 7.1

Install Red Hat Enterprise Linux Server on all the servers that will participate in the cluster. For more information on deploying Red Hat Enterprise Linux Server, visit:

https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Installation_Guide/

4.4. *OpenShift Deployment*

After the Red Hat Enterprise Linux 7 servers have been deployed and all the OpenShift pre-requisites have been met, OpenShift can be deployed onto the servers. The following steps provide direction on how to accomplish this. See Figure 4.1 for an overview of the topology.

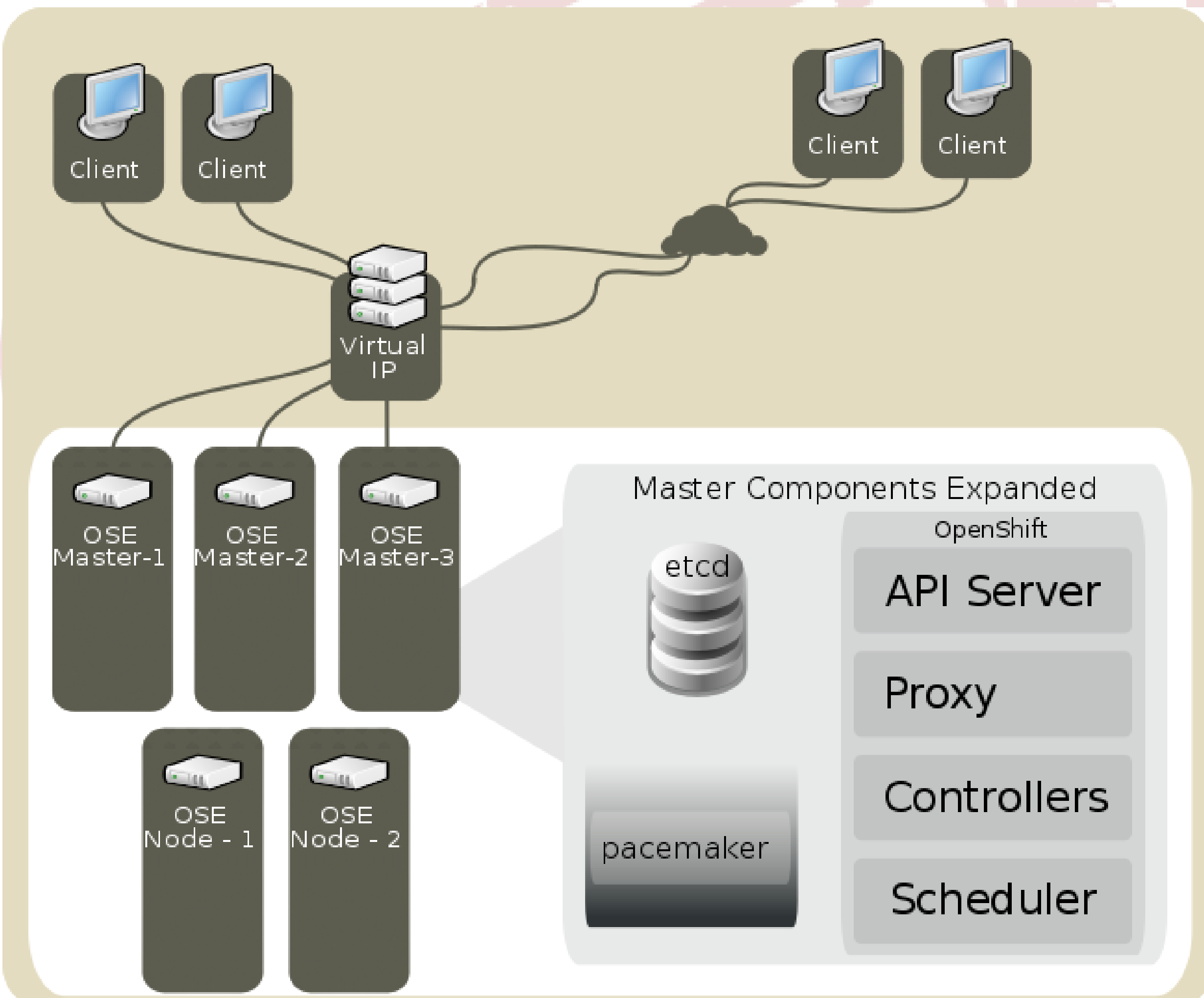

Figure 4.1: OpenShift Highly Available Deployment

The following commands should be executed on the server labeled `master-1` unless otherwise specified.

1. Register the system with Red Hat Network in order to ensure access to entitled content.

```
# subscription-manager register --username=UsernameHere --password=hunter2 --auto
-attach --force
WARNING

This system has already been registered with Red Hat using RHN Classic.

Your system is being registered again using Red Hat Subscription Management. Red Hat
recommends that customers only register once.

To learn how to unregister from either service please consult this Knowledge Base
Article: https://access.redhat.com/kb/docs/DOC-45563
The system has been registered with ID: 2d7dxx45-fbeb-43cc-9ecb-89cxx83cb955

Installed Product Current Status:
Product Name: Red Hat Enterprise Linux Server
Status:       Subscribed
```

2. Subscribe to the HA pool and the OSEv3 pool.

```
# subscription-manager subscribe --pool <pool id> --pool <pool id>
Successfully attached a subscription for: OpenShift Enterprise Broker Infrastructure
Successfully attached a subscription for: High-Availability (8 sockets)

# subscription-manager repos --enable rhel-7-server-ose-3.0-rpms
Repository 'rhel-7-server-ose-3.0-rpms' is enabled for this system.
```

3. Disable all the repositories.

```
# subscription-manager repos --disable="*"
Repository 'rh-gluster-3-nagios-for-rhel-7-server-debug-rpms' is disabled for this
system.
Repository 'rhel-7-server-ose-3.0-source-rpms' is disabled for this system.
Repository 'rhel-7-server-v2vwin-1-debug-rpms' is disabled for this system.

[ ... Output Abbreviated ... ]

Repository 'rh-gluster-3-splunk-for-rhel-7-server-rpms' is disabled for this system.
Repository 'rhel-7-server-rt-beta-rpms' is disabled for this system.
Repository 'rhel-7-server-rhceph-1.3-mon-debug-rpms' is disabled for this system.
```

4. Enable the appropriate repositories for the that are required for the master nodes.

```
# subscription-manager repos --enable="rhel-7-server-rpms" --enable="rhel-7-server
-extras-rpms" --enable="rhel-7-server-ose-3.0-rpms" --enable="rhel-ha-for-rhel-7
-server-rpms"
Repository 'rhel-7-server-rpms' is enabled for this system.
Repository 'rhel-7-server-optional-rpms' is enabled for this system.
Repository 'rhel-ha-for-rhel-7-server-rpms' is enabled for this system.
Repository 'rhel-7-server-extras-rpms' is enabled for this system.
Repository 'rhel-7-server-ose-3.0-rpms' is enabled for this system.
```

Run the same command on the nodes except remove the high availability channel from the list as it is not required on the nodes.

Perform the following steps on master-1 or the system that you are running Ansible on:

1. Install the EPEL repository. Installing EPEL can also be done on a separate host that has access to the three master servers.

```
# yum -y install https://dl.fedoraproject.org/pub/epel/7/x86_64/e/epel-release-7-
5.noarch.rpm
Loaded plugins: langpacks, product-id, rhnplugin, subscription-manager
This system is not registered with RHN Classic or Red Hat Satellite.
You can use rhn_register to register.
Red Hat Satellite or RHN Classic support will be disabled.
epel-release-7-5.noarch.rpm
|  14 kB  00:00:00
Examining /var/tmp/yum-root-jXHkck/epel-release-7-5.noarch.rpm: epel-release-7-
5.noarch

[ ... Output Abbreviated ... ]

Running transaction
  Installing : epel-release-7-5.noarch
1/1
  Verifying  : epel-release-7-5.noarch
1/1

Installed:
  epel-release.noarch 0:7-5

Complete!
```

2. Disable the EPEL repository and install Ansible.

```
# yum-config-manager --disable epel
# yum -y --enablerepo=epel install ansible
```

3. Clone the OpenShift install repository and change to the proper directory.

```
# cd
# git clone https://github.com/openshift/openshift-ansible --branch v3.0.1-1
Cloning into 'openshift-ansible'...
remote: Counting objects: 5432, done.
remote: Compressing objects: 100% (26/26), done.
remote: Total 5432 (delta 7), reused 0 (delta 0), pack-reused 5404
Receiving objects: 100% (5432/5432), 1.04 MiB | 0 bytes/s, done.
Resolving deltas: 100% (2673/2673), done.
# cd openshift-ansible
```

4. Configure SSH so that the OpenShift installer can reach each master and perform the required actions.

```
# ssh-keygen -b 2048 -t rsa -f /root/.ssh/id_rsa -q -N ""
# for i in master-{1..3}; do ssh-copy-id -i ~/.ssh/id_rsa.pub $i:; done
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out
any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now
it is to install the new keys
Warning: Permanently added 'master-1,10.x.x.130' (ECDSA) to the list of known hosts.
root@master-1's password:

Number of key(s) added: 1

[ ... Output Abbreviated ... ]

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'master-3'"
and check to make sure that only the key(s) you wanted were added.
```

5. Now repeat the same process to copy the keys to the nodes.

```
# for i in node-{1..2}; do ssh-copy-id -i ~/.ssh/id_rsa.pub $i:; done
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out
any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now
it is to install the new keys
Warning: Permanently added 'node-1,10.x.x.133' (ECDSA) to the list of known hosts.
root@node-1's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'node-1'"
and check to make sure that only the key(s) you wanted were added.

/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out
any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now
it is to install the new keys
Warning: Permanently added 'node-2,10.x.x.134' (ECDSA) to the list of known hosts.
root@node-2's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'node-2'"
and check to make sure that only the key(s) you wanted were added.
```

6. Run the OpenShift installer based on the Ansible playbook. See the note below about the /etc/ansible/hosts file.

```
# ansible-playbook ~/openshift-ansible/playbooks/byo/config.yml
PLAY [Populate config host groups] *

TASK: [fail ]
skipping: .localhost]

TASK: [fail ] *
skipping: .localhost]

TASK: [fail ] *
skipping: .localhost]

TASK: [Evaluate oo_etcd_to_config]
ok: .localhost] => (item=master-1.cloud.example.com)
ok: .localhost] => (item=master-2.cloud.example.com)
ok: .localhost] => (item=master-3.cloud.example.com)

[ ... Output Abbreviated ... ]

PLAY RECAP
localhost                  : ok=12   changed=0    unreachable=0    failed=0
master-1.cloud.example.com : ok=212  changed=66   unreachable=0    failed=0
master-2.cloud.example.com : ok=95   changed=33   unreachable=0    failed=0
master-3.cloud.example.com : ok=95   changed=33   unreachable=0    failed=0
node-1.cloud.example.com : ok=44   changed=16   unreachable=0    failed=0
node-2.cloud.example.com : ok=44   changed=16   unreachable=0    failed=0
```

A copy of a valid inventory file for this deployment can be found in Appendix C. The example inventory file should replace the existing /etc/ansible/hosts file.

4.5. Configure Fencing

Fencing is a mechanism which protects your data from being corrupted by rogue nodes or concurrent access. The process by which this protection happens is called STONITH (Shoot the Other Node in the Head). During the installation of OpenShift Enterprise, STONITH is disabled. The reason for this is because fencing is a unique attribute applied to each environment. In this section fencing will be enabled and configured. The pre-requisites for configuring fencing in this environment are that each iDRAC has an IP Address, IPMI over LAN has been enabled, and authentication has been set up on the iDRAC. These requirements may be different for each hardware vendor, so check with your vendor on how to configure the hardware.

1. Enable STONITH.

```
# pcs property set stonith-enabled=true
```

2. Install the fencing packages.

```
# yum install fence-agents-ipmilan.x86_64
Loaded plugins: langpacks, product-id, rhnplugin, subscription-manager
This system is receiving updates from RHN Classic or Red Hat Satellite.
Resolving Dependencies
--> Running transaction check

[ ... Output Abbreviated ... ]

Installed:
  fence-agents-ipmilan.x86_64 0:4.0.11-13.el7_1.1

Dependency Installed:
  OpenIPMI-modalias.x86_64 0:2.0.19-11.el7     fence-agents-common.x86_64 0:4.0.11-
13.el7_1.1     ipmitool.x86_64 0:1.8.13-8.el7_1
  pexpect.noarch 0:2.3-11.el7

Complete!
```

3. Create the fence devices and ensure the password is set.

```
# pcs stonith create node1-idrac fence_ipmilan pcmk_host_list="master-
1.cloud.example.com" ipaddr="10.x.x.82" login="root" passwd='PASSWORD' --force
# pcs stonith create node2-idrac fence_ipmilan pcmk_host_list="master-
2.cloud.example.com" ipaddr="10.x.x.83" login="root" passwd='PASSWORD' --force
# pcs stonith create node3-idrac fence_ipmilan pcmk_host_list="master-
3.cloud.example.com" ipaddr="10.x.x.84" login="root" passwd='PASSWORD' --force

# pcs stonith update node1-idrac password= passwd='PASSWORD' --force
# pcs stonith update node2-idrac password= passwd='PASSWORD' --force
# pcs stonith update node3-idrac password= passwd='PASSWORD' --force
```

4. List the fence devices.

```
# pcs stonith show
 node1-idrac    (stonith:fence_ipmilan):    Started
 node2-idrac    (stonith:fence_ipmilan):    Started
 node3-idrac    (stonith:fence_ipmilan):    Started
```

For more information on fencing check the Red Hat High Availibility documentation here: https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Configuring_the_Red_Hat_High_Availability_Add-On_with_Pacemaker/ch-fencing-HAAR.html

At this point, the installation is complete.

5. Operational Management

With the successful deployment of OpenShift, the following section demonstrates how to confirm proper functionality of the Red Hat High Availability Add-On.

5.1. *Pacemaker*

The following steps are to be performed on the master-1 server unless otherwise specified.

1. List pacemakers status

```
# systemctl status pacemaker
pacemaker.service - Pacemaker High Availability Cluster Manager
   Loaded: loaded (/usr/lib/systemd/system/pacemaker.service; enabled)
   Active: active (running) since Mon 2015-08-17 20:34:44 EDT; 34min ago
 Main PID: 2362 (pacemakerd)
   CGroup: /system.slice/pacemaker.service
            ├─2362 /usr/sbin/pacemakerd -f
            ├─2562 /usr/libexec/pacemaker/cib
            ├─2565 /usr/libexec/pacemaker/stonithd
            ├─2566 /usr/libexec/pacemaker/lrmd
            ├─2567 /usr/libexec/pacemaker/attrd
            ├─2569 /usr/libexec/pacemaker/pengine
            └─2572 /usr/libexec/pacemaker/crmd

Aug 17 20:34:45 master-1.cloud.example.com crmd[2572]: notice: crm_update_peer_state:
pcmk_quorum_notification: Node master-1.cloud...(null))
Aug 17 20:34:45 master-1.cloud.example.com crmd[2572]: notice: crm_update_peer_state:
pcmk_quorum_notification: Node master-2.cloud...(null))

[ ... Output Abbreviated ... ]

Aug 17 20:34:45 master-1.cloud.example.com crmd[2572]: notice: do_started: The local
CRM is operational
Aug 17 20:35:02 master-1.cloud.example.com crmd[2572]: notice: process_lrm_event:
Operation virtual-ip_monitor_0: not running (node=maste...d=true)
Hint: Some lines were ellipsized, use -l to show in full.
```

2. List cluster status.

```
# pcs status
Cluster name: openshift_master
Last updated: Sat Sep  5 09:17:36 2015      Last change: Fri Sep  4 17:44:54 2015 by
root via crm_resource on master-1.cloud.example.com
Stack: corosync
Current DC: master-2.cloud.example.com (version 1.1.13-a14efad) - partition with
quorum
3 nodes and 5 resources configured

Online: [ master-1.cloud.example.com master-2.cloud.example.com master-
3.cloud.example.com ]

Full list of resources:

 Resource Group: openshift-master
     virtual-ip (ocf::heartbeat:IPaddr2):   Started master-2.cloud.example.com
     master (systemd:openshift-master): Started master-2.cloud.example.com
 node1-idrac    (stonith:fence_ipmilan):    Started master-1.cloud.example.com
 node2-idrac    (stonith:fence_ipmilan):    Started master-1.cloud.example.com
 node3-idrac    (stonith:fence_ipmilan):    Started master-1.cloud.example.com

PCSD Status:
  master-1.cloud.example.com: Online
  master-2.cloud.example.com: Online
  master-3.cloud.example.com: Online

Daemon Status:
  corosync: active/enabled
  pacemaker: active/enabled
  pcsd: active/enabled
```

The **virtual-ip** resource is located on master-2.cloud.example.com

3. List pacemaker resources.

```
# pcs resource show
 Resource Group: openshift-master
     virtual-ip (ocf::heartbeat:IPaddr2):   Started
     master (systemd:openshift-master): Started
```

4. Place master-2 in standby and get status.

```
# pcs cluster standby master-2.cloud.example.com
# pcs status
Cluster name: openshift_master
Last updated: Sat Sep  5 09:19:07 2015      Last change: Sat Sep  5 09:18:45 2015 by
root via crm_attribute on master-1.cloud.example.com
Stack: corosync
Current DC: master-2.cloud.example.com (version 1.1.13-a14efad) - partition with
quorum
3 nodes and 5 resources configured

Node master-2.cloud.example.com: standby
Online: [ master-1.cloud.example.com master-3.cloud.example.com ]

Full list of resources:

 Resource Group: openshift-master
     virtual-ip (ocf::heartbeat:IPaddr2):   Started master-1.cloud.example.com
     master (systemd:openshift-master): Started master-1.cloud.example.com
 node1-idrac    (stonith:fence_ipmilan):    Started master-1.cloud.example.com
 node2-idrac    (stonith:fence_ipmilan):    Started master-1.cloud.example.com
 node3-idrac    (stonith:fence_ipmilan):    Started master-1.cloud.example.com

PCSD Status:
  master-1.cloud.example.com: Online
  master-2.cloud.example.com: Online
  master-3.cloud.example.com: Online

Daemon Status:
  corosync: active/enabled
  pacemaker: active/enabled
  pcsd: active/enabled
```

Notice how master-2 is listed as online but it is in standby mode now.

5. Set master-2 out of standby state and check the status.

```
# pcs cluster unstandby master-2.cloud.example.com
# pcs status

[ ... Output Abbreviated ... ]
```

6. Verify the default cluster configuration set by the OpenShift Installer.

```
# pcs config
Cluster Name: openshift_master
Corosync Nodes:
 master-1.cloud.example.com master-2.cloud.example.com master-3.cloud.example.com
Pacemaker Nodes:
 master-1.cloud.example.com master-2.cloud.example.com master-3.cloud.example.com

Resources:
 Group: openshift-master
  Resource: virtual-ip (class=ocf provider=heartbeat type=IPaddr2)
   Attributes: ip=10.x.x.15
   Operations: start interval=0s timeout=20s (virtual-ip-start-timeout-20s)
               stop interval=0s timeout=20s (virtual-ip-stop-timeout-20s)
               monitor interval=10s timeout=20s (virtual-ip-monitor-interval-10s)
  Resource: master (class=systemd type=openshift-master)
   Operations: start interval=0s timeout=90s (master-start-timeout-90s)
               stop interval=0s timeout=90s (master-stop-timeout-90s)
               monitor interval=60s (master-monitor-interval-60s)

Stonith Devices:
 Resource: node1-idrac (class=stonith type=fence_ipmilan)
  Attributes: pcmk_host_list=master-1.cloud.example.com ipaddr=10.x.x.82 login=root
passwd=100Mgmt- password= lanplus=1
  Operations: monitor interval=60s (node1-idrac-monitor-interval-60s)
 Resource: node2-idrac (class=stonith type=fence_ipmilan)
  Attributes: pcmk_host_list=master-2.cloud.example.com ipaddr=10.x.x.83 login=root
passwd=100Mgmt- password= lanplus=1
  Operations: monitor interval=60s (node2-idrac-monitor-interval-60s)
 Resource: node3-idrac (class=stonith type=fence_ipmilan)
  Attributes: pcmk_host_list=master-3.cloud.example.com ipaddr=10.x.x.84 login=root
passwd=100Mgmt- password= lanplus=1
  Operations: monitor interval=60s (node3-idrac-monitor-interval-60s)
Fencing Levels:

Location Constraints:
Ordering Constraints:
Colocation Constraints:

Cluster Properties:
 cluster-infrastructure: corosync
 cluster-name: openshift_master
 dc-version: 1.1.13-a14efad
 have-watchdog: false
 stonith-enabled: true
```

7. Additional commands can be performed on the cluster. The following commands provide more information about the environment.

```
# pcs cluster
# pcs cluster report
# pcs resource
etc...
```

8. If a node needs to be shut down immediately and taken out of the cluster, it can be fenced.

```
# pcs stonith fence master-3.cloud.example.com --off
```

5.2. *OpenShift*

The following section describes functionality and usability of the virtual IP address.

1. Verify functionality of the openshift-master service which runs on one master node.

```
# systemctl status openshift-master
openshift-master.service - Cluster Controlled openshift-master
   Loaded: loaded (/usr/lib/systemd/system/openshift-master.service; disabled)
  Drop-In: /run/systemd/system/openshift-master.service.d
           └─50-pacemaker.conf
   Active: active (running) since Mon 2015-08-17 20:35:09 EDT; 39min ago
     Docs: https://github.com/openshift/origin
 Main PID: 2909 (openshift)
   CGroup: /system.slice/openshift-master.service
           └─2909 /usr/bin/openshift start master
--config=/etc/openshift/master/master-config.yaml --loglevel=4

Aug 17 21:14:03 master-2.cloud.example.com openshift-master[2909]: I0817
21:14:03.414473    2909 nodecontroller.go:314] node node-2.cloud...

[ ... Output Abbreviated ... ]

Aug 17 21:14:08 master-2.cloud.example.com openshift-master[2909]: I0817
21:14:08.578687    2909 nodecontroller.go:314] node node-1.cloud...
Hint: Some lines were ellipsized, use -l to show in full.
```

2. Prior to checking virtual-ip resource connectivity, ensure the VIP resides on master-1.

```
# ip a s em1
2: em1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP qlen 1000
    link/ether 44:a8:42:af:57:fe brd ff:ff:ff:ff:ff:ff
    inet 10.x.x.130/21 brd 10.19.143.255 scope global dynamic em1
       valid_lft 3167sec preferred_lft 3167sec
    inet 10.x.x.15/21 brd 10.19.143.255 scope global secondary em1
       valid_lft forever preferred_lft forever
    inet6 2620:52:0:1388:46a8:42ff:feaf:57fe/64 scope global dynamic
       valid_lft 2591805sec preferred_lft 604605sec
    inet6 fe80::46a8:42ff:feaf:57fe/64 scope link
       valid_lft forever preferred_lft forever
```

The virtual-ip is associated with em1 interface on master-1.

3. Connect to the VIP and ensure that the client traffic is working.

```
# oc --server='https://vip.cloud.example.com:8443' get nodes
NAME                                      LABELS
STATUS
node-1.cloud.example.com   kubernetes.io/hostname=node-1.cloud.example.com   Ready
node-2.cloud.example.com   kubernetes.io/hostname=node-2.cloud.example.com   Ready
```

4. Move the resource to master-2, check the interface on master-2.

```
# pcs resource move virtual-ip master-2.cloud.example.com
# pcs status
# ip a s em1

[ ... Output Abbreviated ... ]
```

5. Once the resource has been successful transferred to master-2, re verify the VIP connection to ensure client traffic is still working.

```
# oc --server='https://vip.cloud.example.com:8443' get nodes
NAME                                      LABELS
STATUS
node-1.cloud.example.com   kubernetes.io/hostname=node-1.cloud.example.com   Ready
node-2.cloud.example.com   kubernetes.io/hostname=node-2.cloud.example.com   Ready
```

5.3. *etcd*

This section focuses on the etcd cluster. It describes the different commands to ensure the cluster is healthy.

1. Issue the `etcdctl` command to confirm that the cluster is healthy.

```
# etcdctl -C https://master-1.cloud.example.com:2379,https://master-
2.cloud.example.com:2379,https://master-3.cloud.example.com:2379 --ca-file
/etc/etcd/ca.crt --cert-file=/etc/openshift/master/master.etcd-client.crt --key
-file=/etc/openshift/master/master.etcd-client.key cluster-health
cluster is healthy
member 33831519f1425cdb is healthy
member 5a2b972b868f9514 is healthy
member c421184c33f2c077 is healthy
```

In this configuration the `etcd` services are distributed among the OpenShift master nodes. An alternative configuration would be to distribute the `etcd` service amongst its own nodes. An example /etc/ansible/hosts file showing that configuration can be found in Appendix D. The etcd cluster requires three additional nodes. The only change in the file is the hostnames in the [etcd] stanza.

6. Conclusion

Red Hat solutions involving the OpenShift PaaS are created to deliver a production-ready foundation that simplifies the deployment process, shares the latest best practices, and provides a stable highly available environment on which to run your production applications.

A successful deployment consists of the following:

- Properly configured servers that meet the OpenShift pre-requisites.
- Properly configured DNS.
- Access to a host with Ansible support.
- Proper subscription entitlement on all servers.

Once these requirements are met, the highly available OpenShift environment can be deployed.

For any questions or concerns, please email refarch-feedback@redhat.com and ensure to visit the Red Hat Reference Architecture page to find about all of our Red Hat solution offerings.

DRAFT

Appendix A: Revision History

Revision	Release Date	Author(s)
1.0	Tuesday September 8, 2015	Scott Collier
PDF generated by Asciidoctor PDF		
Reference Architecture Theme version 1.0		

DRAFT

Appendix B: Contributors

1. Jason DeTiberus, content provider
2. Matthew Farrellee, content provider
3. Tim St. Clair, content provider
4. Rob Rati, content provider
5. Aaron Weitekamp, review
6. James Shubin, review
7. Dirk Herrmann, review
8. John Ruemker, review
9. Andrew Beekhof, review

DRAFT

Appendix C: Ansible Host Configuration File - Shared master and etcd services.

```
# Create an OSEv3 group that contains the masters and nodes groups
[OSEv3:children]
masters
nodes
etcd

# Set variables common for all OSEv3 hosts
[OSEv3:vars]
ansible_ssh_user=root
deployment_type=enterprise

# htpasswd auth
#openshift_master_identity_providers=[{'name': 'htpasswd_auth', 'login': 'true',
'challenge': 'true', 'kind': 'HTPasswdPasswordIdentityProvider', 'filename':
'/etc/openshift/htpasswd'}]

# host group for masters

openshift_master_cluster_password=openshift_cluster
openshift_master_cluster_vip=10.x.x.15
openshift_master_cluster_public_vip=10.x.x.15
openshift_master_cluster_hostname=vip.cloud.example.com
openshift_master_cluster_public_hostname=vip.cloud.example.com

[masters]
master-1.cloud.example.com
master-2.cloud.example.com
master-3.cloud.example.com

[etcd]
master-1.cloud.example.com
master-2.cloud.example.com
master-3.cloud.example.com

# host group for nodes
[nodes]
node-[1:2].cloud.example.com openshift_node_labels="{'region': 'primary', 'zone':
'default'}"
```

Appendix D: Ansible Host Configuration File - Dedicated master and dedicated etcd services.

```
# Create an OSEv3 group that contains the masters and nodes groups
[OSEv3:children]
masters
nodes
etcd

# Set variables common for all OSEv3 hosts
[OSEv3:vars]
ansible_ssh_user=root
deployment_type=enterprise

# htpasswd auth
#openshift_master_identity_providers=[{'name': 'htpasswd_auth', 'login': 'true',
'challenge': 'true', 'kind': 'HTPasswdPasswordIdentityProvider', 'filename':
'/etc/openshift/htpasswd'}]

# host group for masters

openshift_master_cluster_password=openshift_cluster
openshift_master_cluster_vip=10.x.x.15
openshift_master_cluster_public_vip=10.x.x.15
openshift_master_cluster_hostname=vip.cloud.example.com
openshift_master_cluster_public_hostname=vip.cloud.example.com

[masters]
master-1.cloud.example.com
master-2.cloud.example.com
master-3.cloud.example.com

[etcd]
etcd-1.cloud.example.com
etcd-2.cloud.example.com
etcd-3.cloud.example.com

# host group for nodes
[nodes]
node-[1:2].cloud.example.com openshift_node_labels="{'region': 'primary', 'zone':
'default'}"
```

DRAFT