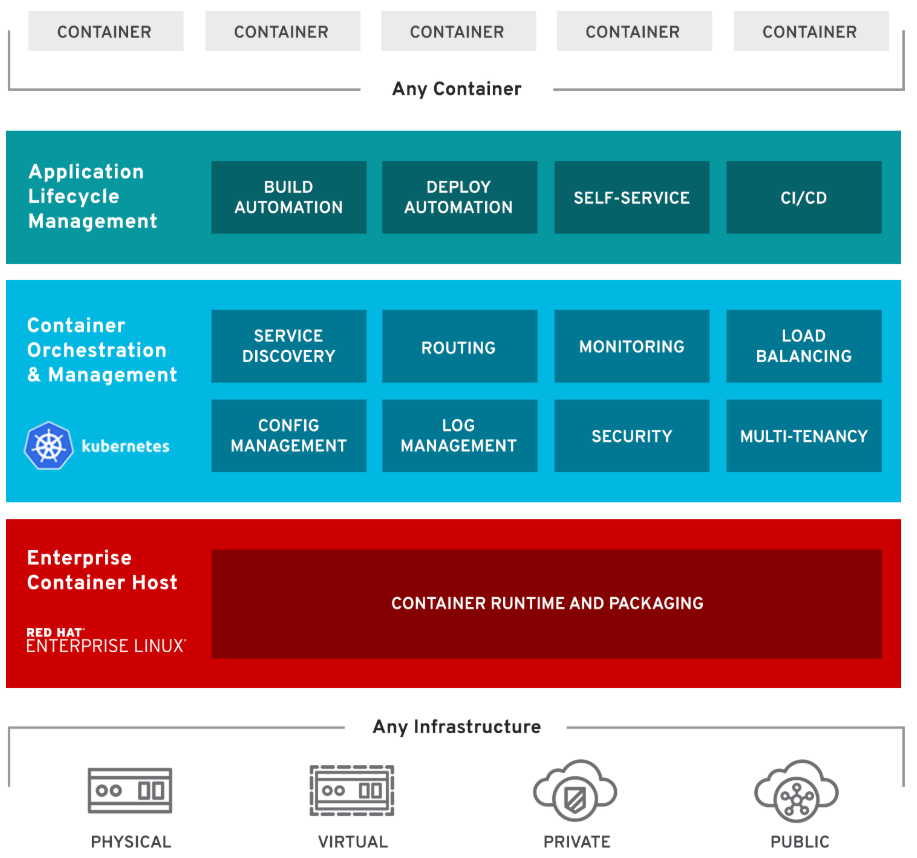
# Deploying OpenShift OKD Cluster setup

# OpenShift Layered Structure­



OpenShift Features

1. **Traditional and Cloud-Native**: Handles cloud-native and traditional applications on a single platform
2. **Multi-tenancy**: Gives your teams self-service to consistent environments across your organization, from development through production.
3. **Hybrid Infra Support**: Offers trusted, proven Kubernetes on any infrastructure.
4. **Security**: Integrated throughout OpenShift, from infrastructure to services, and throughout the operations and application lifecycle
5. **Service on demand**: Seamlessly configure, deploy and consume private and public cloud services with OpenShift Service Catalog.
6. **Built-in Automation**: Remove routine tasks from application teams to focus on business value.
7. **Pluggable Architecture**: Your choice of supported, container runtimes, networking, storage, CI/CD solutions.
8. OpenShift Components:

# OpenShift Components:

Server roles

There are two types of roles a server can play in OpenShift:

* **Master**: Holding the key services, managing nodes, load distribution, management of OpenShift platform, orchestration. Typically, a master server has the API server, etcd, controller manager, and HAProxy (load balance)
* **Node**: Handles the actual load, act as compute provider. Provides runtime environment for containers. Typically, a node has docker service, kubelet, network proxy services.
* **ETCD**: The **openshift** Master, **openshift** provides a REST endpoint for interacting with the system. An **etcd** server, **Openshift** uses **etcd** to store system configuration and state. Controllers: Controllers are the components that run with the masters that makes sure the running system matches the desired state as stored in **etcd**.

## Services

**API Service**:

* Validates and configures the data for pods, services, and replication controllers.
* Synchronizes pod information with service configuration

**Etcd service**: Stores the persistent master state

**Controller Manager service**: Watches etcd for changes to replication controller objects and then uses the API to enforce the desired state.

**HAProxy service**: Provides load balancing for master servers.

**Kubelet**: Updates the node as specified by a container manifest

**Network proxy**: Reflects the services defined in the API on that node.

## Important Components

**OCR**: Provides users with a built-in location for their application builds to push the resulting images

**Web console**: A user interface accessible from a web browser

**OC CLI**: A CLI-based interface to manage OpenShift and use services.

**Pod**:

* Pods are the rough equivalent of a machine instance (physical or virtual) to a container.
* Each pod is allocated its own internal IP address, therefore owning its entire port space.
* Containers within pods can share their local storage and networking.
* Pod is a smallest compute unit that can be defined, deployed, and managed.

## Users

Users are those who usage and consumed the services.

1. **Regular users** can login and can management their applications as per need.
2. **System users** are created automatically for the internal services usage, authentication and platform management.

## Setup Requirements

1. Operating System: RHEL 7.6
2. Master: 2core and 4 GB
3. Nodes: Three nodes with 2 core and 4GB RAM.
4. Domain name: sndoscluster.hobbylobby.corp, \*.apps. sndoscluster.hobbylobby.corp
5. Private IP Scheme: Any private scheme except 172.16.x.x/16
6. Connectivity: Private IP connectivity with password-less SSH access from master to all servers.

Note: Whatever commands and values are mentioned with italic and in yellow color need to be replaced with your own requirements.

## Multiple Masters Using Native HA with External Clustered etcd

The following describes an environment for three masters, one HAProxy load balancer, three external clustered etcd hosts, and two nodes using the **native** HA method:

| **Host Name** | **Infrastructure Component to Install** |
| --- | --- |
| **sndosmasterxx.hobbylobby.corp** | Master (clustered using native HA) and node |
| **sndosmasterxx.hobbylobby.corp** |
| **sndosmasterxx.hobbylobby.corp** |
| **sndoslbxx.hobbylobby.corp** | HAProxy to load balance API master endpoints |
| **sndosinfraxx.hobbylobby.corp** | Clustered etcd |
| **sndosinfraxx.hobbylobby.corp** |
| **sndosinfraxx.hobbylobby.corp** |
| **sndospodxx.hobbylobby.corp** | Node |
| **sndospodxx.hobbylobby.corp** |

## Only on Master server or Ansible Admin server

The following steps only need to be performed on your server you deiced to act as ansible deployment server.

1. Installing ansible on master server using the following command.

yum -y --enablerepo=epel install ansible pyOpenSSL

1. Create the project directory using the following command.

*mkdir openshift*

*cd openshift*

1. Cloning the OpenShift ansible git repository using the following command.

git clone https://github.com/openshift/openshift-ansible.git

1. Check out the desired stable version ( e.g v3.11) using the following command.

cd openshift-ansible && git fetch && git checkout release-3.11 && cd ..

## On all master and nodes servers

OpenShift OKD needs a working DNS server that can resolve the master and node servers FQDNs. If your OpenShift OKD also need to be accessed from public network, then you must have proper domain name registered with the following records:

<Public ip of master server> <record type> <public domain name>

<x.x.x.x> A <master.domian.com>

<x.x.x.x> A <console.domian.com>

<x.x.x.x> A <\*.domian.com>

<x.x.x.x> A <\*apps.domian.com>

<x.x.x.x> A <node01.domian.com>

<x.x.x.x> A <node02.domian.com>

Alternatively, you can add all host entries in host file of all servers. However, \*.domain.com does not work for host file so either you must have proper DNS domain name, or you must register each project’s public domain name in your host file.

1. Make the appropriate changes in the /etc/hosts file on all servers.

vi /etc/hosts

*xx.xxx.xx.xx sndosmasterxx.hobbylobby.corp sndosmasterxx*

*xx.xxx.xx.xx Sndosmasterxx.hobbylobby.corp Sndosmasterxx*

*xx.xxx.xx.xx Sndosmasterxx.hobbylobby.corp Sndosmasterxx*

*xx.xxx.xx.xx Sndosinfraxx.hobbylobby.corp Sndosinfraxx*

*xx.xxx.xx.xx Sndosinfraxx.hobbylobby.corp Sndosinfraxx*

*xx.xxx.xx.xx Sndosinfraxx.hobbylobby.corp Sndosinfraxx*

*xx.xxx.xx.xx Sndoslbxx.hobbylobby.corp Sndoslbxx*

*xx.xxx.xx.xx Sndospodxx.hobbylobby.corp Sndospodxx*

*xx.xxx.xx.xx Sndospodxx.hobbylobby.corp Sndospodxx*

Note: You need to replace the appropriate IPs and Domain names as per your setup.

## Only on master server.

1. Restart and enable the docker service on all servers

systemctl stop docker && systemctl restart docker && systemctl enable docker

1. Generating the ssh keys and coping the public key to all servers for ansible connection from master.

ssh-keygen

Distribute the key to the other cluster hosts. You can use a bash loop:

for host in *xxx.xxx.xxx.xxx* \

*xxx.xxx.xxx.xxx* \

*xxx.xxx.xxx.xxx*; \

*xxx.xxx.xxx.xxx*; \

*xxx.xxx.xxx.xxx*; \

*xxx.xxx.xxx.xxx*; \

*xxx.xxx.xxx.xxx*; \

*xxx.xxx.xxx.xxx*; \

*xxx.xxx.xxx.xxx*; \

do ssh-copy-id -i ~/.ssh/id\_rsa.pub $host; \

done

\* Confirm that you can access each host that is listed in the loop through SSH.

Registering hosts

To access the installation packages, you must register each host with Red Hat Subscription Manager (RHSM) and attach an active OpenShift Container Platform subscription.

1. On each host, register with RHSM:

# subscription-manager register --username=<user\_name> --password=<password>

1. Pull the latest subscription data from RHSM:

# subscription-manager refresh

1. List the available subscriptions:

# subscription-manager list --available --matches '\*OpenShift\*'

1. In the output for the previous command, find the pool ID for an OpenShift Container Platform subscription and attach it:

# subscription-manager attach --pool=<pool\_id>

1. Disable all yum repositories:
   * 1. Disable all the enabled RHSM repositories:

# subscription-manager repos --disable="\*"

* + 1. List the remaining yum repositories and note their names under repo id, if any:

# yum repolist

* + 1. Use yum-config-manager to disable the remaining yum repositories:

# yum-config-manager –disable <repo\_id>

Alternatively, disable all repositories:

# yum config-manager –disable \\*

1. Enable only the repositories required by OpenShift Container Platform 3.11.

* For cloud installations and on-premise installations on x86\_64 servers, run the following command:

# subscription-manager repos \

--enable="rhel-7-server-rpms" \

--enable="rhel-7-server-extras-rpms" \

--enable="rhel-7-server-ose-3.11-rpms" \

--enable="rhel-7-server-ansible-2.6-rpms"

## Installing base packages On master and all nodes servers

Once the setup is ready with basic requirements, proceed for the below steps. The following commands need to be executed on all master and node servers included in your cluster.

1. Switch to root user using the following command.

sudo su -

cd

1. Install the prerequisites with basic tools using the following command.

yum install -y wget git zile nano net-tools docker-1.13.1 bind-utils iptables-services bridge-utils bash-completion kexec-tools sos psacct openssl-devel httpd-tools NetworkManager python-cryptography python2-pip python-devel python-passlib java-1.8.0-openjdk-headless "@Development Tools"

1. Update the yum packages using the following command.

yum update

reboot

1. Install packages that are required for your installation method:

* We are planning to use the RPM-based installer, install the following package:

# yum install openshift-ansible

* Ensure the host is up to date by upgrading to the latest Atomic tree if one is available:

# atomic host upgrade

* After the upgrade is completed and prepared for the next boot, reboot the host:

# reboot

Installing Docker

At this point, install Docker on all master and node hosts. This allows you to configure your Docker storage options before you install OpenShift Container Platform.

\* The cluster installation process automatically modifies the **/etc/sysconfig/docker** file.

1. Install Docker 1.13:

# yum install docker-1.13.1

1. Verify that version 1.13 was installed:

# rpm -V docker-1.13.1

# docker version

Configuring Docker Storage

You must configure storage for all master and node hosts because by default each system runs a container daemon. For containerized installations, you need storage on masters. Also, by default, the web console and etcd, which require storage, run in containers on masters. Containers run on nodes, so storage is always required on them

|  |  |
| --- | --- |
|  |  |

1. Creating an inventory file for ansible playbook, paste the content of inventory.ini file in this file.

vi inventory

The sample of inventory file can be found here.

<https://github.com/vipin-k/Openshift-Cluster/blob/master/inventory.ini>

1. Execute the prerequisites ansible playbook using the following command.

cd openshift

ansible-playbook -i /opt/openshift/inventory /usr/share/ansible/openshift-ansible/playbooks/prerequisites.yml

Note: Make sure the ansible user mentioned in the inventory file is able to access all master and node servers before to execute playbook.

1. Execute the OpenShift cluster deploy playbook using the following command.

ansible-playbook -i /opt/openshift/inventory /usr/share/ansible/openshift- ansible/playbooks/deploy\_cluster.yml

Note: in inventory file you have two options: either have two node including master acting as one node, or have three sperate nodes if you don’t consider master as node.

1. If you get an error due to previous attempt, you can run the following command to remove the existing setup.

ansible-playbook -i inventory.ini openshift-ansible/playbooks/adhoc/uninstall.yml

1. Generating the OpenShift OKD admin password using the following command.

htpasswd -c /etc/origin/master/htpasswd admin

1. Login to **oc cluster** with the system admin user using the following command.

oc login -u system:admin

If you get oc command not found, then you may need to set the path using below command “export PATH=/usr/local/bin/”

1. Adding admin user as cluster admin role using the following command.

oc adm policy add-cluster-role-to-user cluster-admin admin

1. Listing the current nodes in cluster using the following command.

oc get nodes

1. Inspect the list of existing pods in the project by using the following command**.**

oc get pods -o wide

1. Listing the current projects using the following command.

oc projects

1. Login to web console using the following web link.

*https://sndoscluster.hobbylobby.corp:8443*

Note: Make sure you open the port 8443 on your server, network and firewall. Also replace the appropriate domain URL registered for as per your setup.

You should be able to login to OpenShift OKD cluster console. Now, you can proceed to explore the OpenShift OKD platform for creation of projects, building, deploying and scaling your apps. For more details how to use OpenShift OKD platform, please visit the following link and explore the functions, features, and options of OpenShift OKD platform.

<https://learn.openshift.com>