OpenShift 3.10 on VxFlex Ready Node 14th Generation Servers

Abstract
This white paper provides guidance on deployment and exercising basic functionality of Red Hat® OpenShift® Container Platform on Dell EMC VxFlex Ready Nodes for customers requiring an on-prem container platform solution.

March 2019
## Revisions

<table>
<thead>
<tr>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>March 2019</td>
<td>Initial release</td>
</tr>
</tbody>
</table>

The information in this publication is provided “as is.” Dell Inc. makes no representations or warranties of any kind with respect to the information in this publication, and specifically disclaims implied warranties of merchantability or fitness for a particular purpose.

Use, copying, and distribution of any software described in this publication requires an applicable software license.

Copyright © 2019 Dell Inc. or its subsidiaries. All Rights Reserved. Dell, EMC, Dell EMC and other trademarks are trademarks of Dell Inc. or its subsidiaries. Other trademarks may be trademarks of their respective owners. [3/18/2019] [White Paper] [000048]
Table of contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Revisions</td>
<td>2</td>
</tr>
<tr>
<td>Table of contents</td>
<td>3</td>
</tr>
<tr>
<td>Executive summary</td>
<td>5</td>
</tr>
<tr>
<td>1 Introduction</td>
<td>6</td>
</tr>
<tr>
<td>1.1 Audience</td>
<td>6</td>
</tr>
<tr>
<td>1.2 Terminology</td>
<td>6</td>
</tr>
<tr>
<td>2 Product overview</td>
<td>7</td>
</tr>
<tr>
<td>2.1 VxFlex Ready Node</td>
<td>7</td>
</tr>
<tr>
<td>2.2 VxFlex OS</td>
<td>7</td>
</tr>
<tr>
<td>2.3 OpenShift Container Platform</td>
<td>8</td>
</tr>
<tr>
<td>2.3.1 Containers and Docker Engine</td>
<td>8</td>
</tr>
<tr>
<td>2.3.2 Kubernetes</td>
<td>8</td>
</tr>
<tr>
<td>2.3.3 Ansible</td>
<td>9</td>
</tr>
<tr>
<td>3 Architecture</td>
<td>10</td>
</tr>
<tr>
<td>3.1 Logical design</td>
<td>10</td>
</tr>
<tr>
<td>3.2 Virtual machine design</td>
<td>11</td>
</tr>
<tr>
<td>3.3 Configuring VxFlex OS storage</td>
<td>12</td>
</tr>
<tr>
<td>3.3.1 Configuring Storage Pools and Volumes</td>
<td>12</td>
</tr>
<tr>
<td>3.4 Physical design</td>
<td>14</td>
</tr>
<tr>
<td>4 OpenShift Container Platform deployment</td>
<td>16</td>
</tr>
<tr>
<td>4.1 Prerequisites</td>
<td>16</td>
</tr>
<tr>
<td>4.2 Attaching the OpenShift Container Platform subscription</td>
<td>16</td>
</tr>
<tr>
<td>4.3 Setting up repositories</td>
<td>17</td>
</tr>
<tr>
<td>4.4 Install the OpenShift Container Platform package</td>
<td>17</td>
</tr>
<tr>
<td>4.5 Set up password-less SSH access</td>
<td>17</td>
</tr>
<tr>
<td>4.6 Run the installation playbooks</td>
<td>18</td>
</tr>
<tr>
<td>4.7 Understanding roles and authentication</td>
<td>19</td>
</tr>
<tr>
<td>5 POD deployment on an OpenShift Container Platform cluster</td>
<td>20</td>
</tr>
<tr>
<td>5.1 Verifying the installation</td>
<td>20</td>
</tr>
<tr>
<td>5.2 Pods and services</td>
<td>21</td>
</tr>
<tr>
<td>5.2.1 Pods</td>
<td>21</td>
</tr>
<tr>
<td>5.2.2 Service</td>
<td>21</td>
</tr>
<tr>
<td>5.3 Creating a new application</td>
<td>22</td>
</tr>
<tr>
<td>5.4 Deploying applications using the web console</td>
<td>22</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>Conclusion</td>
<td>25</td>
</tr>
<tr>
<td>Technical support and resources</td>
<td>26</td>
</tr>
</tbody>
</table>
Executive summary

Dell EMC VxFlex Ready Nodes help reduce time-to-value for customers who are building a hyperconverged environment using VxFlex OS. These pre-validated, tested, and optimized server nodes help deploy an enterprise-ready VxFlex OS based software-defined storage infrastructure.

Containers are an abstraction at the application layer (sometimes called “operating system virtualization”), which packages software into a standard lightweight format (for example, Docker) with everything an application requires to run successfully (source code, runtime, system tools, system libraries, environment variables, and so on) making it extremely portable. Containers can help lower the barrier of entry to developing microservice based applications allowing developers to spend less time worrying about runtimes, dependencies and differences between test/dev and production environments and more time innovating to meet the demands of today’s dynamic business environment. Container Orchestrators (COs) like Kubernetes make running containers at production scale possible by handling the complexity of managing hundreds or thousands of containers at any one time.

Red Hat® OpenShift® Container Platform offers enterprises full control over their Kubernetes environments, whether they are on-premise or in the public cloud, giving teams the freedom to build and run applications anywhere.

This white paper provides guidance on deployment and exercising basic functionality of Red Hat® OpenShift® Container Platform on Dell EMC VxFlex Ready Nodes for customers requiring an on-prem container platform solution to meet their needs.
1 Introduction

1.1 Audience
The audience for this paper includes sales engineers, field consultants, IT administrators, customers, and anyone else interested in configuring and deploying OpenShift on Dell EMC VxFlex Ready Nodes with VxFlex OS as the underlying software defined storage later.

Readers are expected to have an understanding and working knowledge of containers, Kubernetes, OpenShift, Ansible, VxFlex OS, RHEL, VxFlex Ready Nodes, and iDRAC.

1.2 Terminology
The following table lists the terminology and acronyms that are used throughout this document:

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDC</td>
<td>Storage Data Client</td>
</tr>
<tr>
<td>SDS</td>
<td>Storage Data Server</td>
</tr>
<tr>
<td>MDM</td>
<td>Metadata Manager</td>
</tr>
<tr>
<td>OS</td>
<td>Operating System</td>
</tr>
<tr>
<td>RHEL</td>
<td>RedHat Enterprise Linux</td>
</tr>
<tr>
<td>LAN</td>
<td>Local Area Network</td>
</tr>
<tr>
<td>SAN</td>
<td>Storage Area Network</td>
</tr>
<tr>
<td>HDD</td>
<td>Hard Disk Drive</td>
</tr>
<tr>
<td>SSD</td>
<td>Solid State Drive</td>
</tr>
<tr>
<td>PCIe</td>
<td>Peripheral Component Interconnect Express</td>
</tr>
<tr>
<td>DNS</td>
<td>Domain Name System</td>
</tr>
<tr>
<td>DHCP</td>
<td>Dynamic Host Configuration Protocol</td>
</tr>
<tr>
<td>OCP</td>
<td>OpenShift Container Platform</td>
</tr>
</tbody>
</table>
Product overview

The chapter provides an overview of the different products used in this solution. For more details on these products see the Technical support and resources section.

2.1 VxFlex Ready Node

Dell EMC VxFlex Ready Nodes bring together Dell EMC PowerEdge servers with Dell EMC VxFlex OS storage software in scalable, reliable and easy-to-deploy building blocks for hyper-converged or server SAN architecture, multi-hypervisor or bare metal environments, and high-performance databases. Dell EMC VxFlex Ready Nodes converge storage and compute resources, aggregating capacity and performance with simplified management capable of starting small and scaling in discrete increments. Customers can take advantage of the scalability of VxFlex OS on a rack-optimized PowerEdge server designed to reduce the time spent in planning and deploying new infrastructure.

Dell EMC VxFlex Ready Nodes are tuned and optimized for VxFlex OS. This includes checking that all components follow Dell EMC approved manufacturers’ lists (AMLs) for firmware, BIOS versions, hardware compatibility list (HCL) lookup and driver downloads. Therefore, customers deploying a hyper-converged or two-layer system can depend on this optimized building block to provide the resiliency and availability for their business demands.

2.2 VxFlex OS

VxFlex OS, based on the VxFlex OS software platform, is a software-defined storage solution that uses x86 servers' local disks and LAN to create a SAN that has all the benefits of external storage—but at a fraction of the cost and complexity. VxFlex OS uses local storage devices and turns them into shared block storage. For many workloads, VxFlex OS storage is comparable to, or better than external shared block storage.

The lightweight VxFlex OS software components are installed on the application servers and communicate using a standard LAN to handle the application I/O requests sent to VxFlex OS block volumes. An extremely efficient decentralized block I/O flow, combined with a distributed, sliced volume layout, results in a massively parallel I/O system that can scale up to thousands of nodes.

VxFlex OS is designed and implemented with enterprise-grade resilience. Furthermore, the software features an efficient distributed self-healing process that overcomes media and server failures, without requiring administrator involvement.

Dynamic and elastic, VxFlex OS enables administrators to add or remove servers and capacity on-the-fly. The software immediately responds to the changes, rebalancing the storage distribution and achieving a layout that optimally suits the new configuration.

VxFlex OS software works efficiently with various types of disks, including: magnetic (HDD) and solid-state disks (SSD), flash PCI Express (PCIe) cards, networks, and hosts.

The VxFlex OS SAN consists of the following software components:

**Storage Data Client (SDC):** The SDC is a lightweight, block device-driver that presents VxFlex OS shared block volumes to applications that resides on the same server on which the SDC is installed.

**Storage Data Server (SDS):** The SDS manages the local storage that contributes to the VxFlex OS storage pools. The SDS runs on each of the servers that contribute storage to the VxFlex OS system. The SDS performs the back-end operations that SDCs request.
MetaData Manager (MDM): The MDM configures and monitors the VxFlex OS components. It contains all the metadata required for VxFlex OS operations. You can configure MDM in redundant cluster mode, with three members on three servers or five members on five servers, or in single mode on a single server.

Note: Dell EMC does not recommend using single mode in production systems, except in temporary situations. The MDMs contains all the metadata required for system operation. Single mode has no protection and exposes the system to a single point of failure.

Note: For more information on VxFlex OS, see: https://support.emc.com/products/33925_VxFlex-OS/Documentation/

2.3 OpenShift Container Platform

Red Hat® OpenShift® Container Platform (OCP) is a container application platform that brings Docker and Kubernetes to the enterprise. It provides enterprise-grade Kubernetes environments for building, deploying, and managing container-based applications across any public or private datacenter where Red Hat Enterprise Linux is supported.

Note: For more information, see: https://www.redhat.com/en/technologies/cloud-computing/openshift

2.3.1 Containers and Docker Engine

A container is a standard unit of software that packages up code and all its dependencies, so the application runs quickly and reliably from one computing environment to another.

A Docker container image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings.

Note: For more information, see: https://www.docker.com/resources/what-container

2.3.2 Kubernetes

Kubernetes is a portable, extensible open-source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation. It has a large, rapidly growing ecosystem. Kubernetes services, support, and tools are widely available.

Note: For more information, see: https://kubernetes.io/ and https://www.redhat.com/en/topics/containers/what-is-kubernetes.

2.3.2.1 Storage for Kubernetes

OCP uses the Kubernetes persistent volume (PV) framework to allow cluster administrators to provision persistent storage for a cluster manually or dynamically. Developers can use persistent volume claims (PVCs) to request PV resources without having specific knowledge of the underlying storage infrastructure. A Kubernetes StorageClass provides a way for administrators to describe the “classes” of storage they offer. Different classes might map to quality-of-service levels, or to backup policies, or to arbitrary policies determined by the cluster administrators. Kubernetes itself is unopinionated about what classes represent.
Container Storage Interface (CSI) is a community driven standard for persistent storage on container orchestrators (COs) like Kubernetes. It enables storage providers to develop a single CSI driver for any CO that has implemented CSI. The Kubernetes implementation of CSI was promoted to GA in Kubernetes 1.13 and is compatible with CSI spec v1.0.

**Note:** For more information about CSI with Kubernetes, see https://kubernetes-csi.github.io/docs/Home.html.

OCP 3.10 supports the following Persistent Volume plugins: NFS, HostPath, GlusterFS, Ceph RBD, OpenStack Cinder, AWS Elastic Block Store (EBS), GCE Persistent Disk, iSCSI, Fibre Channel, Azure Disk, Azure File, VMWare vSphere and Local.

**Note:** CSI volumes are currently in Technology Preview for OCP 3.10 and not for production workloads. CSI volumes may change in a future release of OpenShift Container Platform. Technology Preview features are not supported with Red Hat production service level agreements (SLAs), might not be functionally complete, and Red Hat® does not recommend using them for production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process. See the Technology Preview features support scope for more information: https://access.redhat.com/support/offerings/techpreview/?extIdCarryOver=true&sc_cid=701f2000001Css5AAC.

Future updates to this document will occur with subsequent OCP versions, which offer GA support of CSI to highlight VxFlex OS CSI driver usage for PVs.

### 2.3.3 Ansible

Ansible is a simple open source IT engine, which automates application deployment, intra service orchestration, cloud provisioning and many other IT tools. Ansible is easy to deploy because it does not use any agents or custom security infrastructure.

Ansible uses playbooks to describe automation jobs using a very simple language (for example, human-readable data serialization language like YAML). Ansible is designed for multi-tier deployment. It does not manage one system at a time, it models IT infrastructure by describing all your systems as interrelated.

**Note:** For more information, see: https://docs.ansible.com/
### 3 Architecture

In this chapter, we provide an overview of the logical design, VM design, and the network design which was used in this deployment.

#### 3.1 Logical design

Figure 1 shows a brief list of all the components used in our OCP deployment on VxFlex Ready Nodes based on VxFlex Ready Node R640s.

![Logical design using VxFlex OS 2.6.1 and RHEL 7.5 KVM on Dell EMC VxFlex Ready Node R640s](image-url)
3.2 Virtual machine design

Once the VxFlex OS components are installed, proceed with VxFlex OS deployment. Later, we can install VxFlex OS GUI server to configure the previously installed SDS and SDC.

We can carve out new volumes and assign to all the hosts and consume these volumes for our OCP VMs.

We created three VMs running RedHat® Enterprise Linux in each of the hosts including:

- Openshift® Master VM
- Openshift® Infrastructure VM
- Openshift® App VM
3.3 Configuring VxFlex OS storage

Before we begin our Red Hat® OpenShift® deployment, we need to check for all the prerequisites and create a dedicated Bastion (Management) node with RHEL OS installed on it. This Bastion node hosts DNS, DHCP services, VxFlex OS Gateway Server, and VxFlex OS GUI. We also use this host for deploying Ansible and we will be running all our playbooks from this host for Red Hat® OpenShift® deployments.

The VxFlex Ready Nodes based on PowerEdge R640 that are part of our Red Hat® OpenShift® cluster are updated with appropriate BIOS and firmware versions. RedHat® Enterprise Linux is installed on all the VxFlex Ready Nodes that will participate in the VxFlex OS cluster. Additional KVM packages and other libraries are also installed on these hosts for enabling virtualization. With virtualization enabled we can configure high availability for OpenShift masters and infrastructure nodes in an environment where physical nodes are limited in number.

Appropriate VxFlex OS packages for RedHat® Enterprise Linux are uploaded to VxFlex OS Gateway server. After the upload is completed using VxFlex Installation Manager, we installed all the VxFlex OS components including: MDM, SDS, and SDC on each of the VxFlex Ready Nodes.

After installing VxFlex OS, we must configure the VxFlex OS cluster. To accomplish that, we should first install VxFlex OS GUI client on a Microsoft Windows or Linux machine that has access to the VxFlex OS nodes and then connect to the primary MDM IP address. Once logged in, we must accept the secondary MDM certificate and license.

The Protection Domain is configured by default during cluster creation, but storage pools and volumes still need to be configured. Create a storage pool and add all the drives from each of the VxFlex OS clustered nodes. Next, set the data path IPs and virtual IP for the cluster in the server GUI.

**Note:** For more details on VxFlexOS installation download the VxFlex OS documentation set: [https://support.emc.com/docu89115_VxFlex_Software_Only_2.6_Documentation_Set.zip?language=en_US&language=en_US](https://support.emc.com/docu89115_VxFlex_Software_Only_2.6_Documentation_Set.zip?language=en_US&language=en_US)

3.3.1 Configuring Storage Pools and Volumes

VxFlex OS is used as infrastructure storage on top of which the OCP is deployed. After the VxFlex OS cluster deployment completes, you must perform following tasks to configure infrastructure storage for deploying OCP

- Associate VxFlex OS storage pool with physical disks
- Create and map VxFlex OS volumes

**Before you begin**

You must issue these commands from the Master MDM node, either directly, or using SSH. Ensure that you have the authentication credentials. In addition, you can use the VxFlex OS GUI. The VxFlex OS CLI (SCLI) is installed as part of the MDM component. After the VxFlex OS system is installed, follow these steps to configure local storage.
Adding SDS devices

- Add at least one device to at least three SDSs, with a minimum of 100 GB free storage capacity per device.
- Balance the total device capacity over all SDSs.

1. Log in to the VxFlex OS cluster:

   scli --login --username <MDM_USERNAME> --password <MDM_PASSWORD>

   **Note:** If bash completion is not enabled, run: /etc/bash_completion.d/scli.

2. If one does not already exist, define a Protection Domain, `pd1`, by running the following command from the Master MDM (MDM 1):

   scli --add_protection_domain --protection_domain_name pd1

3. Add a Storage Pool, `sp1`, by running the following command:

   scli --add_storage_pool --protection_domain_name pd1 --storage_pool_name sp1

4. Add devices using the following command:

   scli --add_sds_device --sds_ip <IP> --protection_domain_name <NAME> --storage_pool_name <NAME> --device_path <DEVICE_PATH>

   **Example:**

   scli --add_sds_device --sds_ip 192.168.212.10 -pd1 default -sp1 default --device_path /dev/sdX

5. Add a volume:

   scli --add_volume --protection_domain_name <NAME> --storage_pool_name <NAME> --size_gb <SIZE> --volume_name <NAME>

   **Example 1:**

   scli --add_volume --protection_domain_name default --storage_pool_name default --size_gb 16 --volume_name vol01

6. Map a volume to an SDC:

   scli --map_volume_to_sdc --volume_name <NAME> --sdc_ip <IP>

   **Example 2:**

   scli --map_volume_to_sdc --volume_name vol01 --sdc_ip 192.168.212.19
**Note:** In Linux, mapped volumes appear to the SDC as `/dev/sciniX` where `X` is a letter, starting from "a". For more information, see "Mounting VxFlex OS" in the VxFlex OS documentation.

You can use the VxFlex OS GUI or the `SCLI --query_all` command to see the installed nodes and storage.

Next, you can start using your storage. For best results, Dell EMC recommends reviewing the VxFlex OS User Guide.

**Note:** For full description please check - [https://support.emc.com/products/33925_VxFlex-OS/Documentation/](https://support.emc.com/products/33925_VxFlex-OS/Documentation/)

### 3.4 Physical design

In Figure 3, each VxFlex Ready Node has two dual port ethernet adapters. Therefore, four ethernet ports per node and are configured for the following traffic types: RHEL/KVM Bridge network, VxFlex OS management traffic, VxFlex Data 1 and VxFlex Data 2 traffic respectively on each of these interfaces.

![Physical design diagram](image)

**Figure 3** Shows physical and logical networking topology for deploying Red Hat® OpenShift® on VxFlex Ready Nodes R640 servers.

**Note:** For a production environment, you must adhere to recommendations with NIC redundancy as per the VxFlex OS best practices.

Two Top of Rack (TOR) switches were used for redundancy and load balancing purposes. We also configured VPC on both the TOR switches, which in a production environment would be used for end-user facing traffic. Finally, we created appropriate VLANs to separate different traffic types on the physical interfaces.
Note: For more detailed network configuration information, see the *VxFlex Ready Node Hardware Configuration and Operating System Installation Guide*, which is available from the *VxFlex Ready Node (no AMS) Documentation Library*. And also the networking best practice guide: [https://www.emc.com/collateral/white-papers/h17332-dellmc-vxflex-os-networking-best-practices.pdf](https://www.emc.com/collateral/white-papers/h17332-dellmc-vxflex-os-networking-best-practices.pdf).
OpenShift Container Platform deployment

4 OpenShift Container Platform deployment

This chapter introduces you to the basic concepts of OCP and helps you install a basic application using infrastructure storage from VxFlex OS. The purpose of this document is to outline the basic constructs of an OCP deployment on VxFlex Ready Nodes.

4.1 Prerequisites

To install OCP, you need:

- At least three physical or virtual RHEL 7+ machines, with fully qualified domain names (either real world or within a network) and password-less SSH access to each other. This guide uses master.openshift.example.com and node.openshift.example.com. These machines must ping each other using these domain names.
- A valid Red Hat subscription.
- Wildcard DNS resolution that resolves your domain to the IP of the node. An entry like the following in your DNS server:

  master.flex.com. 300 IN A <master_ip>
  node.flex.example.com. 300 IN A <node_ip>
  *.apps.flex.example.com. 300 IN A <node_ip>

You might wonder why are the apps in your domain name used for the wildcard entry? When using OCP to deploy applications, an internal router must proxy incoming requests to the corresponding application pod. By using apps as part of the application domains, the application traffic is accurately marked to the right pod. This domain can be set to any desired value. In the example below, we use the cloudapps domain.

  *.cloudapps.flex.com. 300 IN A <node_ip>

Once these prerequisites are configured, you must set up an OCP install.

4.2 Attaching the OpenShift Container Platform subscription

1. As user root on the target machines (Master, App and Infra node), use subscription-manager commands at the Linux prompt to register the systems with Red Hat.

   # subscription-manager register

2. Pull the latest subscription data from Red Hat Subscription Manager (RHSM):

   # subscription-manager refresh

3. List the available subscriptions.

   # subscription-manager list --available

4. Find the pool ID that provides the OpenShift Container Platform subscription and attach it.

   # subscription-manager attach --pool=<pool_id>

5. Replace the string <pool_id> with the pool ID of the pool that provides the OpenShift Container Platform. The pool ID is a long alphanumeric string.
These RHEL systems are now authorized to install OCP. Next, you must tell the systems where to get OCP.

4.3 Setting up repositories

On all nodes, use subscription-manager to enable the necessary repositories for installing OCP. You may have already enabled the first two repositories in this example.

1. For cloud installations and on-premise installations on x86_64 servers, run the following command in the Linux prompt on each node:

```
# subscription-manager repos --enable="rhel-7-server-rpms" \
--enable="rhel-7-server-extras-rpms" \
--enable="rhel-7-server-ose-3.10-rpms" \
--enable="rhel-7-server-ansible-2.4-rpms"
```

This command tells your RHEL system that the tools required to install OCP are available from these repositories. Next, we need the OCP installer that is based on Ansible.

4.4 Install the OpenShift Container Platform package

The installer for OCP is provided by the `openshift-ansible` package. Install it using the `yum` command on the Master after running `yum update`. YUM (Yellowdog Updater Modified) is part of RedHat Package Manager based Linux systems and is an open source command-line and graphical based package management tool.

```
# yum -y install wget git net-tools bind-utils iptables-services bridge-utils
bash-completion kexec-tools sos psacct
# yum -y update
# reboot
# yum -y install openshift-ansible
```

Now, select and install a container engine:

- To install CRI-O:
  
  ```
  # yum -y install cri-o
  ```

- To install Docker:
  
  ```
  # yum -y install docker
  ```

4.5 Set up password-less SSH access

1. Before running the installer on the Master, set up password-less SSH access. This is required by the installer to gain access to the machines. On the Master, run the following command in the Linux prompt.

   ```
   # ssh-keygen
   ```

2. Follow the prompts and press Enter when asked for pass phrase.
3. An easy way to distribute your SSH keys is by using a bash loop:

   ```
   # for host in master.openshift.example.com \
   ```
4.6 Run the installation playbooks

This section describes how to run the installation playbooks.

Note: Before you can run the installation playbooks, ensure you have configured VxFlex OS storage as described in Section 3.3 Configuring VxFlex OS Storage.

1. Log in to the Master node where the openshift-ansible package is installed.

   Note: After installing openshift-ansible package, you can find the hosts file populated in the default location.

2. Edit the example inventory to use your host names, and then save it to a file. The default location is /etc/ansible/hosts.

   Example:

   ```
   # cat /etc/ansible/singlehosts.example
   ```

   ```
   [root@host123 ansible]# cat singlehosts.example
   # Creates an OSEv3 group that contains the masters, nodes, and etc groups
   [OSEv3:children]
   masters
   nodes
   etc
   # Set variables common for all OSEv3 hosts
   [OSEv3:vars]
   # SSH user, this user should allow ssh based auth without requiring a password
   ansible_ssh_user=root
   # If ansible_ssh_user is not root, ansible_become must be set to true
   ansible_become=true
   openshift_deployment_type=openshift-enterprise
   oreg_url=registry.access.redhat.com/openshift3/ose-#{component};#{version}
   openshift_examples_modify_imagestream=true
   # Uncomment the following to enable https authentication, defaults to DenyAllPasswordIdentityProvider
   #openshift_master_identity_providers=[
   # 'name': 'https Pswd auth', 'login': 'true', 'challenge': 'true', 'kind': 'HTTPPasswordIdentityProvider'
   #]
   # host group for masters
   [masters]
   host123.flex.com
   # host group for etcd
   [etcd]
   host123.flex.com
   # host group for nodes, includes region info
   [nodes]
   host123.flex.com openshift_node_group_name='node-config-master'
   host18.flex.com openshift_node_group_name='node-config-compute'
   infra-node1.example.com openshift_node_group_name='node-config-compute'
   infra-node2.example.com openshift_node_group_name='node-config-compute'
   ```

   Figure 4 Example of user defined host inventory file

3. Run the prerequisites.yml playbook using your inventory file:
OpenShift Container Platform deployment

$ ansible-playbook -i <inventory_file> /usr/share/ansible/openshift-ansible/playbooks/prerequisites.yml

4. Run the deploy_cluster.yml playbook using your inventory file:

$ ansible-playbook -i <inventory_file> /usr/share/ansible/openshift-ansible/playbooks/deploy_cluster.yml

After a successful install, but before you add a new project, you must set up basic authentication, user access, and routes.

4.7 Understanding roles and authentication

By default, when installed for the first time, there are no roles or user accounts created in OCP, so you must create them. You have the option to either create new roles or define a policy that allows anyone to log in (to start you off).

1. Before you do anything else, log in at least one time with the default system:admin user.

2. On the Master, run the following command in Linux prompt:

   $ oc login -u system:admin

   **Note:** All commands must be run on the Master, unless otherwise indicated.

   By logging in at least one time with this account, you create the system:admin user’s configuration file, which allows you to log in subsequently.

   There is no password for this system account.

3. Run the following command to verify that the OpenShift Container Platform was installed and started successfully. You will get a listing of the master and node, in the Ready Node status.

   $ oc get nodes

   **Note:** For full description, see:
   [https://docs.openshift.com/container-platform/3.10/cli_reference/get_started_cli.html](https://docs.openshift.com/container-platform/3.10/cli_reference/get_started_cli.html)
5 POD deployment on an OpenShift Container Platform cluster

After OCP cluster is deployed successfully, you must verify the nodes.

5.1 Verifying the installation

1. After the installation completes, log in to the Master node using the default admin credentials:

   ```
   # oc login
   Username: system:admin
   ```

2. Switch to the default created project.

   ```
   # oc project default
   ```

3. Verify that the Master has started, and that the nodes are registered and reporting in Ready Node status. On the master host, run the following as root:

   ```
   [root@master1 ~]# oc get nodes
   NAME               STATUS    ROLES     AGE       VERSION
   app1.flex.com      Ready     compute   58d       v1.10.0+b81c8f8
   app2.flex.com      Ready     compute   58d       v1.10.0+b81c8f8
   app3.flex.com      Ready     compute   58d       v1.10.0+b81c8f8
   infra1.flex.com    Ready     infra     58d       v1.10.0+b81c8f8
   infra2.flex.com    Ready     infra     58d       v1.10.0+b81c8f8
   infra3.flex.com    Ready     infra     58d       v1.10.0+b81c8f8
   master1.flex.com   Ready     master    58d       v1.10.0+b81c8f8
   master2.flex.com   Ready     master    58d       v1.10.0+b81c8f8
   master3.flex.com   Ready     master    58d       v1.10.0+b81c8f8
   ```

4. To verify that the web console is installed correctly, use the Master host name and the web console port number to access the web console with a web browser.

   For example, for a master host with a host name of master.openshift.com and using the default port of 8443, the web console would be found at `https://master.flex.com:8443/console`

---

**Note:** Log in with admin and the https password if you set in inventory file `/etc/ansible/hosts`

After your log in, the web page looks like Figure 5.
5.2 Pods and services

5.2.1 Pods
Pods are defined as a collection of containers and their storage inside a node of OpenShift (Kubernetes) cluster. In general, we have two types of pods starting from a single container pod to multi-container pod.

**Note:** For more information, see: [https://kubernetes.io/docs/concepts/workloads/pods/pod/](https://kubernetes.io/docs/concepts/workloads/pods/pod/)

5.2.2 Service
As we have a set of containers running inside a pod, in the same way we have a service that is defined as a logical set of pods. A service is an abstracted layer on top of the pod that provides a single IP and DNS name through which pods are accessed. The services help in managing the load balancing configuration and to easily scale the pods. In OCP, a service is a REST object whose identification is posted to `apiService` on OpenShift Master to create a new instance.

**Note:** For more information, see
[https://docs.openshift.com/container-platform/3.9/architecture/core_concepts/pods_and_services.html](https://docs.openshift.com/container-platform/3.9/architecture/core_concepts/pods_and_services.html)
[https://kubernetes.io/docs/concepts/services-networking/service/](https://kubernetes.io/docs/concepts/services-networking/service/)
5.3 Creating a new application
In this section, we deploy the Tomcat application on our OCP cluster.

Using CLI:
1. Log in to the Master node and run the following commands:

   # oc new-app tomcat

2. After this is successful, check the pods.

   # oc get pods
   tomcat-1-6t7xp 1/1 Running 0 23d

3. Log in to the pod using the CLI:

   # oc rsh <pod_name>

Note: For more information on how to use Openshift CLI, see: https://docs.openshift.com/container-platform/3.10/cli_reference/get_started_cli.html

5.4 Deploying applications using the web console
This section shows you how to deploy applications using the web console.

1. Select the application you want to deploy (For example: .NET CORE).

   ![Application catalog](image)

   Figure 6  Application catalog

2. Click Next.
3. Update all your details, select your project, version required, application name and repository link from where you must deploy the image and then click **Create**.
4. After it successfully completes, click Close.

![Test Core application deployment results](image1)

**Figure 9** .NET Core application deployment results

5. Check the application deployed in your dashboard.

![Test Core application after successful deployment](image2)

**Figure 10** .NET Core application after successful deployment

**Note**: Sample Git Repository location was used in above example.
6 Conclusion

Red Hat® OpenShift® Container Platform is a complete container application platform that natively integrates technologies like Docker and Kubernetes—a powerful container cluster management and orchestration system—and includes an enterprise foundation in Red Hat Enterprise Linux®.

Dell EMC VxFlex Ready Nodes converge storage and compute resources, aggregating capacity and performance with simplified management capable of starting small and scaling in discrete increments. Dell EMC VxFlex Ready Nodes bring together industry leading PowerEdge servers with Dell EMC VxFlex OS software in scalable, reliable and easy-to-deploy building blocks for hyper-converged or server SAN architecture, multi-hypervisor or bare metal environments, and high-performance databases. In addition, it offers flexibility in deployment options.

This White paper outlined the physical and logical architectural guidelines for deploying and using the Red Hat® OpenShift® Container Platform on a minimal cluster of VxFlex OS using Dell EMC VxFlex Ready Nodes for customers requiring an on-prem container platform solution to meet their needs.
Technical support and resources

See the following referenced or recommended resources related to this document:

- VxFlex Ready Node
- VxFlex Ready Node (no AMS) Documentation Library
- Red Hat® OpenShift® Container Platform 3.10
- Install OpenShift Container Platform
- VxFlex OS Documentation
- DELL EMC FLEX FAMILY AND VXFLEX OS: THE SOFTWARE BEHIND THE SYSTEMS