Red Hat OpenShift 4.2 on Dell EMC VxFlex Ready Nodes

Installation and Configuration overview with VxFlex Ready Nodes

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-premises container platform solution.

March 2020
Revisions

<table>
<thead>
<tr>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>March 2020</td>
<td>Initial release</td>
</tr>
</tbody>
</table>

Acknowledgements

This paper was produced by the following:

Author: Sunil Kumar HS and Raghavendra Biligiri
Support: Raghvendra Tripathi, David J Adams and Shalini G

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 © 2020 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. [3/27/2020] [White Paper] [000062]
# Table of contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>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</td>
<td>7</td>
</tr>
<tr>
<td>2.1 VxFlex family</td>
<td>7</td>
</tr>
<tr>
<td>2.1.1 VxFlex integrated rack</td>
<td>7</td>
</tr>
<tr>
<td>2.1.2 VxFlex appliance</td>
<td>8</td>
</tr>
<tr>
<td>2.1.3 VxFlex Ready Nodes</td>
<td>8</td>
</tr>
<tr>
<td>2.1.4 VxFlex OS</td>
<td>8</td>
</tr>
<tr>
<td>2.2 Red Hat Virtualization</td>
<td>9</td>
</tr>
<tr>
<td>2.3 Red Hat OpenShift Container Platform 4.2</td>
<td>9</td>
</tr>
<tr>
<td>2.4 Kubernetes</td>
<td>9</td>
</tr>
<tr>
<td>2.4.1 Container Storage Interface</td>
<td>10</td>
</tr>
<tr>
<td>2.4.2 Red Hat Enterprise Linux CoreOS</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>11</td>
</tr>
<tr>
<td>3.1 Logical architecture</td>
<td>12</td>
</tr>
<tr>
<td>3.2 Physical design</td>
<td>13</td>
</tr>
<tr>
<td>3.3 CSAH Node Installation and Configuration</td>
<td>14</td>
</tr>
<tr>
<td>3.4 Configuring VxFlex OS storage</td>
<td>15</td>
</tr>
<tr>
<td>3.4.1 Configuring storage pools and volumes</td>
<td>15</td>
</tr>
<tr>
<td>4</td>
<td>17</td>
</tr>
<tr>
<td>4.1 Deploying OCP cluster</td>
<td>18</td>
</tr>
<tr>
<td>4.2 Adding RHEL 7.6 worker nodes to OCP cluster</td>
<td>20</td>
</tr>
<tr>
<td>4.3 Verifying cluster setup</td>
<td>21</td>
</tr>
<tr>
<td>5</td>
<td>22</td>
</tr>
<tr>
<td>5.1 Deploying Images</td>
<td>22</td>
</tr>
<tr>
<td>5.2 Deploy applications using S2I</td>
<td>22</td>
</tr>
<tr>
<td>5.3 Application routing</td>
<td>23</td>
</tr>
</tbody>
</table>
# Table of contents

6 Conclusion .................................................................................................................................................. 24
A Appendix: Sample file .......................................................................................................................... 25
B Appendix: Technical support and resources ......................................................................................... 26
   B.1 Related resources .............................................................................................................................. 26
Executive summary

Enterprises across the globe are challenged to improve the business processes, develop new capabilities and business models. Enterprises are digitally transforming to meet the business goals and deliver cutting-edge technology, products, and services. Cloud Native architecture enables rapid development and deployment of agile applications at scale and resiliency. One of the most powerful tools in this new Cloud Native architecture is containerization, which has been evolving new workloads and use cases. Both open-source community and software vendors are constantly adding exciting features to container technology to benefit developers and IT.

Cloud Native architecture enables applications to take advantage of containerization, elasticity, resilience, scale, and orchestration. Cloud Native applications, including databases, require supporting tools that have capabilities to deal with scheduling, load balancing, resource monitoring, scaling, and job isolation of these complex environments. Kubernetes is an open-source container orchestration system with capabilities of automatic, scaling, and managing containerized applications.

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

VxFlex family offers key value propositions of traditional and cloud-native production workloads, deployment flexibility, linear scalability, predictable high performance, and enterprise-grade resilience.

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-premises 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, VxFlex OS, Red Hat Enterprise Linux, 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>CSAH</td>
<td>Cluster System Admin Host</td>
</tr>
<tr>
<td>DHCP</td>
<td>Dynamic Host Configuration Protocol</td>
</tr>
<tr>
<td>DNS</td>
<td>Domain Name System</td>
</tr>
<tr>
<td>HDD</td>
<td>Hard Drive</td>
</tr>
<tr>
<td>LAN</td>
<td>Local Area Network</td>
</tr>
<tr>
<td>MDM</td>
<td>Meta Data Manager</td>
</tr>
<tr>
<td>OCP</td>
<td>OpenShift Container Platform</td>
</tr>
<tr>
<td>OS</td>
<td>Operating System</td>
</tr>
<tr>
<td>PCIe</td>
<td>Peripheral Component Interconnect Express</td>
</tr>
<tr>
<td>RHCOS</td>
<td>Red Hat Enterprise Linux CoreOS</td>
</tr>
<tr>
<td>RHEL</td>
<td>Red Hat Enterprise Linux</td>
</tr>
<tr>
<td>RHV</td>
<td>Red Hat Virtualization</td>
</tr>
<tr>
<td>SAN</td>
<td>Storage Area Network</td>
</tr>
<tr>
<td>SDC</td>
<td>Storage Data Client</td>
</tr>
<tr>
<td>SDS</td>
<td>Storage Data Server</td>
</tr>
<tr>
<td>SSD</td>
<td>Solid-State Drive</td>
</tr>
</tbody>
</table>
2 Product overview

2.1 VxFlex family

VxFlex family helps transforming from a traditional three-tier architecture to a modern data center without any trade-offs, meeting business requirements without compromise. The central software layer for VxFlex family is VxFlex OS, scale-out block storage service that enables customers to create a scale-out Server SAN or hyperconverged infrastructure. The VxFlex family currently include VxFlex Ready Nodes, VxFlex appliance, and VxFlex integrated rack.

![VxFlex family diagram](image)

**Figure 1** VxFlex family

2.1.1 VxFlex integrated rack

VxFlex integrated rack is a rack-scale engineered system, with integrated networking, that provides linear scalability and enterprise-grade availability. VxFlex integrated rack is engineered, manufactured, managed, supported, and sustained as one system for single end-to-end lifecycle support.

VxFlex integrated systems create a server-based SAN by combining virtualization software, which is known as VxFlex OS, with Dell EMC PowerEdge servers to deliver flexible, scalable performance and capacity on demand. Local storage resources are combined to create a virtual pool of block storage with varying performance tiers. The VxFlex integrated rack enables you to scale from a small environment to enterprise scale with over a thousand nodes. In addition, it provides enterprise grade data protection, multi-tenant capabilities, and add-on enterprise features such as QoS, thin provisioning, and snapshots. VxFlex systems deliver the performance and time-to-value required to meet the demands of the modern enterprise data center.
2.1.2 **VxFlex appliance**

VxFlex appliance is a preconfigured and validated for fast, easy deployment, VxFlex appliance offers a turnkey experience in an economic form factor. With VxFlex appliance, customers benefit from a smaller starting point, with massive scale potential, without having to compromise on performance and resiliency.

2.1.3 **VxFlex Ready Nodes**

VxFlex Ready Nodes combine Dell EMC PowerEdge servers that are powered by Intel® Xeon® Scalable Processors and VxFlex OS software to create scalable, reliable, and easy-to-deploy building blocks for hyperconverged or server SAN architecture, multi-hypervisor or bare metal environments, and high-performance databases.

2.1.4 **VxFlex OS**

VxFlex OS applies the principles of server virtualization to standard x86 servers with local disks, creating high-performance, sharable pools of block storage. VxFlex OS abstracts the local storage out of each contained within each server, including HDDs, SSDs, and all-flash. VxFlex OS uses three lightweight pieces of software to create, consume, and coordinate the storage layer in VxFlex systems.

VxFlex OS can be deployed in the following ways:

- **Two-layer**: In a two-layer deployment, compute resources exist on one set of nodes, and storage resources exist on another set of nodes.
- **Hyperconverged (HCI)**: The SDCs and SDSs run on the same set of nodes.
- **Hybrid**: VxFlex OS also enables deployments that mix the two-layer and HCI deployments.

In this solution, VxFlex OS is deployed using HCI.
2.2 Red Hat Virtualization

Red Hat® Virtualization® is a virtualization solution from Red Hat. It is derived from the Red Hat Enterprise Linux® kernel, Kernel-based Virtual Machine (KVM) technology and oVirt virtualization management projects.

The Red Hat Virtualization consists of the following components:

1. Red Hat Virtualization Manager
2. Hosts
3. Storage

For complete information about components, see Red Hat Virtualization components.

You can deploy Red Hat Virtualization as Standalone Manager or Self-Hosted Engine. For complete deployment information, see Red Hat Architecture.

2.3 Red Hat OpenShift Container Platform 4.2

Red Hat OpenShift Container Platform provides developers and IT organizations with a hybrid cloud application platform for deploying both new and existing applications on secure, scalable resources with minimal configuration and management overhead. It provides enterprise-grade Kubernetes environments for building, deploying, and managing container-based applications across any on-premises data center where Red Hat Enterprise Linux is supported.

New features and enhancements in Red Hat OpenShift 4.2 are as follows:

- Operators and Operator Lifecycle Management (OLM).
- Enhanced web console.
- Enhanced Container Storage Interface (CSI).
- Red Hat Enterprise Linux CoreOS (RHCOS) has replaced Atomic host and is made mandatory to use for the control plane nodes.
- Red Hat OpenShift Container Platform 4.2 uses Kubernetes 1.14 with CRI-O as container run time replacing Docker from earlier version 3.
- RHEL 7.6 or RHCOS 4.2 can be used for Worker nodes.
- Installation is done using ignition-based deployment replacing Ansible tool.

For complete information about OpenShift 4.2 release, see Release Notes.

2.4 Kubernetes

Kubernetes is an open-source project which is portable and can be extended to manage containerized workloads and services. Thus, Kubernetes facilitates both declarative configuration and automation.

Containers are a good way to bundle and run your applications because it has its own file system, CPU, memory, process space, and so on. As they are decoupled from the underlying infrastructure, they are portable across clouds and operating system distributions. Kubernetes provides you with a framework to run distributed systems resiliently.

For more information about Kubernetes, see Kubernetes documentation.
### 2.4.1 Container Storage Interface

Kubernetes natively offers some solutions to manage storage, however, native storage options also present challenges with the pod portability. CSI solves the challenges with native Kubernetes solutions. It is a community driven standard for persistent storage on container orchestrators (COs) like Kubernetes. It enables storage providers to develop CSI driver for Kubernetes CO systems. It lets you provision storage for pods through a Kubernetes PersistentVolumeClaim (PVC).

For more information about CSI Overview, see [Container Storage Interface (CSI) for Kubernetes GA](#).

### 2.4.2 Red Hat Enterprise Linux CoreOS

Red Hat Enterprise Linux CoreOS (RHCOS) represents the next generation of single-purpose container operating system technology. RHCOS combines the quality standards of Red Hat Enterprise Linux (RHEL) with the automated, remote upgrade features from Container Linux. RHCOS is the only supported operating system for OpenShift Container Platform control plane, or master, machines.

For more information about RHCOS, see [Red Hat Enterprise Linux CoreOS (RHCOS)](#).
3  Architecture

This section provides overview of logical, virtual machine, and physical architecture of VxFlex OS 3.0.1 and RHV on Dell EMC VxFlex R640 Ready Nodes.

**IMPORTANT:** This whitepaper describes a means of deploying Red Hat OpenShift Container Platform 4.2 on the VxFlex hardware and virtualization infrastructure. For this solution, Dell EMC choose to implement these functions as virtual machines. This approach reduces the amount of infrastructure required while introducing enhanced options for high availability and management. Three VxFlex Ready Nodes host the virtual machines for the OCP cluster as shown in Figure 3.

It is recommended that industry best practice should be observed for production deployment of your applications, such as hosting one OpenShift Master and one Worker on each of the three VxFlex Ready Nodes. This ensures that the complete failure of a physical node permits the continued deployment of a fully viable and highly-available OpenShift cluster.

The next version of the White Paper will provide VxFlex OS CSI and Persistent storage related information.
### 3.1 Logical architecture

The following logical architecture figure demonstrates the HCI layer deployment of VxFlex OS where both compute and storage nodes run on Red Hat Virtualization.

---

**Figure 3: Logical design**

RHV 4.2 and RHEL 7.6 was used as only these versions are supported on VxFlex OS 3.0.1 version.

For VxFlex OS cluster, Red Hat Virtualization 4.2 was deployed on three nodes with both SDC and SDS. VxFlex OS volumes were configured from VxFlex OS storage pool and used for OCP cluster.

**Example:** All the Master or Controller VMs from Master VxFlex OS Volume, Worker VMs from Worker VxFlex OS Volume respectively.

NFS Shared storage was used only to deploy RHVM virtual machine and not for deploying OpenShift Cluster. In the next version of White Paper, planning to deploy RHVM on VxFlex OS Storage Cluster.

For OpenShift Container Cluster, Master and Worker VMs are deployed on three nodes. The SDC component is installed on the OpenShift Worker nodes to allow direct mount of VxFlex OS volumes to OpenShift pods.

The VMs were configured such that same type of Master or Worker nodes were not running on the same host.
3.2 Physical design

In the following physical design figure, you can observe that VxFlex Ready Node has two dual port ethernet adapters:

![Physical design diagram](image)

Since each node has two dual port ethernet adapters, four ethernet ports are configured for each node to control the following traffic types:

- RHEL/KVM Bridge network
- VxFlex OS management
- VxFlex Data 1
- VxFlex Data 2

**Note:** For production environment, it is mandatory to adhere to recommendations with NIC redundancy as per the VxFlex OS best practices.

**Network Design:**

- Two TOR switches that are used for redundancy and load-balancing purposes.
- Configured Virtual Link Trunking (VLT) on both the TOR switches, which in a production environment would be used for end-user facing traffic.
- Finally, created appropriate VLANs to separate different traffic types on the physical interfaces.

For more detailed information about network configuration, see [VxFlex Ready Node Documentation Library](#).
### 3.3 CSAH Node Installation and Configuration

CSAH Node is an administration host of the OpenShift Cluster. This node has all the services required for OpenShift cluster deployment. This host is not part of OCP cluster.

In this deployment, Red Hat Enterprise Linux 7.6 was installed on the CSAH node by configuring the following required services for OCP Cluster deployment on the CSAH node:

- DNS Server
- DHCP server
- HTTP server (web Server)
- HA Proxy for Load Balancing
- NFS Storage (Used for only RHVM, Not for OCP cluster)
- Ansible
- VxFlex Gateway
- VxFlex GUI
- VxFlex SDS (optional)

**DNS Server**

Proper DNS setup is imperative for a functioning OpenShift cluster. DNS is used for name resolution (A records), certificate generation (PTR records), and service discovery (SRV records). OpenShift 4 has a concept of a **cluster name** that was incorporated into clusters DNS records. DNS records have CLUSTER NAME.DOMAIN in them. In other words, “cluster name” ends up being part of FQDN.

Install DNS packages on CSAH Node and configure Forward and Reverse zone file.

**Load Balancer**

Red Hat OpenShift suggests using an enterprise-class Layer-4 load balancer, however in this setup uses Software load balancer HAProxy. You require a load balancer to front-end the APIs, both internal and external, and the OpenShift router. Install HAProxy related packages in CSAH node.

You must configure Port 6443 and 22623 to point to the bootstrap and master nodes. You also must configure 80 and 443 to point to the worker nodes.

**Webserver**

A web server is needed in order to hold the ignition configurations and installation images for when you install Red Hat Enterprise Linux CoreOS. Any web server works, if the web server can be reached by the bootstrap, master, and worker nodes during installation.

**DHCP**

It is recommended to use the DHCP server to manage the IP addresses of the nodes for the cluster long term. Ensure that the DHCP server is configured to provide persistent IP addresses and host names to the cluster machines. Using DHCP with IP reservation ensures the IPs does not change on reboots.

For more details, see [User-provisioned DNS requirements](#).
NFS Shared

NFS shared storage was used only to deploy RHVM virtual machine and not used for OpenShift Cluster.

VxFlex OS Gateway

Before deploying VxFlex OS, you must create a VxFlex OS Gateway on a server.

1. Download and extract the complete software for this version from Dell EMC Support Site: https://support.emc.com/products/33925.
2. Copy the Gateway RPM (RHEL) and proceed to install as shown here:

```
#GATEWAY_ADMIN_PASSWORD=<new_GW_admin_password> rpm -U /tmp/EMC-ScaleIO-gateway-3.0-<build>.X.rpm
```

3.4 Configuring VxFlex OS storage

Prerequisites

Ensure VxFlex OS Gateway service is configured and running on CSAH Node.

Procedure

1. Upload OS package to VxFlex OS Gateway server.
2. Install complete VxFlex OS components - MDM, SDS, and SDC on each of the VxFlex Ready Nodes.
3. Configure VxFlex OS cluster:
   a. Install VxFlex OS GUI server on Microsoft Windows or Linux machine that has access to VxFlex OS nodes.
   b. Connect to the primary MDM IP address to log in.
   c. Accept the secondary MDM certificates and license.
4. By default, the Protection Domain is configured.
5. Create a storage pool and add all the drives from each of the VxFlex OS clustered nodes.
6. Set the data path IPs and virtual IP for the cluster in the server UI.

For more details about VxFlex OS installation, download the complete product documentation set from this link:

3.4.1 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, perform the following tasks to configure infrastructure storage for deploying OCP:

- Associate VxFlex OS storage pool with hard drives.
- Create and map VxFlex OS volumes.
- Create individual volumes size as we required for each VM (Master0, 1, 2, Worker 0, 1).
Prerequisites:

- Issue the commands provided in this section from the Master MDM node, either directly, or using SSH.
- Ensure that you have the authentication credentials.
- Installed VxFlex OS CLI (SCLI) as part of the MDM component.

Requirements:

- 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.

Procedure to add SDS devices:

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/d/scli`.

2. Add devices.

   ```
   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 --protection_domain_name default --storage_pool_name default --device_path /dev/sdX
   ```

3. Add a volume.

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

   Example:
   ```
   scli --add_volume --protection_domain_name default --storage_pool_name default --size_gb 16 --volume_name vol01
   ```

4. Map a volume to an SDC.

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

   Example:
   ```
   scli --map_volume_to_sdc --volume_name vol01 --sdc_ip 192.168.212.19
   ```

**Note:** In Linux, which is 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*.

These results can be checked using VxFlex OS GUI or the SCLI `--query_all` command to see the installed nodes and storage.
**OpenShift Container Platform deployment**

The following flowchart, highlights the steps to be performed for deploying OpenShift Container Platform:

**Step 1:** Ensure to complete the pre-requisite of configuring DNS, DHCP, HA Proxy, Http, and Ansible services in CSAH node as mentioned in [CSAH Node Installation and Configuration](#).

**Step 2:** Deploy and configure Bootstrap VM with RHCOS as mentioned in [Deploying OCP cluster](#).

**Step 3:** Deploy and configure Master VMs with RHCOS as mentioned in [Deploying OCP cluster](#).

**Step 4:** Deploy, configure, and add RHEL 7.6 worker nodes to OCP cluster as mentioned in [Adding RHEL 7.6 worker nodes to OCP cluster](#).
4.1 Deploying OCP cluster

To deploy OCP cluster, complete the following steps in CSAH node:

1. Login to CSAH node.
2. Generate SSH private key.
3. Add the SSH key to ssh-agent and installation program to perform installation debugging or disaster recovery on the cluster.
4. Create a directory and download the following software, openshift-install, openshift-client and the pullsecret files from the Red Hat portal by using your Red Hat credentials.
5. Create installation configuration file: install-config.yaml.

   This file is used to define cluster domain, compute, control plane, networking, ssh-key, and pull-secret in this file.

   Note: To know about the install-config.yaml file details, see sample file available in Appendix A.

6. Create Kubernetes manifest.

   Using this manifest file, create Ignition config files: bootstrap.ign, master.ign, and worker.ign.

7. Create RHCOS machines to use by using either the ISO image of RHCOS or use PXE.

   During installation of RHCOS, ignition files are provided as an input. Once the RHCOS installation is completed, system reboots and applies the Ignition config file which was specified at the time of installation. This results in installation of bootstrap, master, and worker nodes with RHCOS.

8. Wait for the bootstrap process to complete to create the OpenShift Container Platform cluster.
9. Monitor the bootstrap process:

   ```
   $ ./openshift-install --dir=<installation_directory> wait-for bootstrap-complete --log-level=info
   ```

   This command succeeds when the Kubernetes API server signals that it has been bootstrapped on the control plane machines.

10. Log in to the OCP cluster and approve any pending certificates signing request (CSRs) which are generated for each machine when they are added to the cluster.
11. Once the control plane machines are initialized, check and configure the Operators so that all the operators are available.

   ```
   $ watch -n5 oc get clusteroperators
   ```
12. Once the operator configuration is completed the cluster installation can be monitored for its completion using the below command:

```bash
$ ./openshift-install --dir=<installation_directory> wait-for install-complete
INFO Waiting up to 30m0s for the cluster at https://api.test.flex.com:6443 to initialize...
INFO Waiting up to 10m0s for the openshift-console route to be created...
INFO Install complete!
INFO To access the cluster as the system:admin user when using 'oc', run 'export KUBECONFIG=/root/test/auth/kubeconfig'
INFO Access the OpenShift web-console here: https://console-openshiftconsole.apps.test.flex.com
INFO Login to the console with user: kubeadmin, password: xxx
```

**Results:** OCP cluster is up and running with Master nodes. VxFlex OS supports only RHEL nodes, so ensure to add RHEL worker nodes to the OCP cluster.

```
[root@bastion helm]# oc get nodes |grep master
master-0.test.flex.com  Ready  master  77d  v1.14.6+9fb2d5cf9
master-1.test.flex.com  Ready  master  77d  v1.14.6+9fb2d5cf9
master-2.test.flex.com  Ready  master  77d  v1.14.6+9fb2d5cf9
```
4.2 Adding RHEL 7.6 worker nodes to OCP cluster

Prerequisites

- Ensure to have installed RHEL 7.6 operating system on Worker nodes.
- Register each host with Red Hat Subscription Manager (RHSM), attach an active OpenShift Container Platform subscription, and enable the required repositories.

Procedure

1. Log in to CSAH node.
2. Extract the pull secret as .txt file and name it as pull-secret.txt.

   ```
   # oc -n openshift-config get -o jsonpath='{.data...}'
   secret pull-secret | base64 -d | jq .
   ```

3. Specify the following parameters in Ansible host inventory files:
   - **User name**: Set username as root.
   - **Ansible_become**: Set ansible_become to True.
   - **Path**:
     - Specify the path of kubeconfig file in the OpenShift cluster.
     - Specify the path file of pull-secret.txt in the image registry.
   - **Domain name**: Provide fully qualified domain name for each host. This name is the hostname that the cluster uses to access the machine, so set the correct public or private name to access the machine.

   Sample code is provided here in /etc/ansible/hosts

   ```
   [all:vars] ansible_user=root
   #ansible_become=True
   openshift_kubeconfig_path="~/.kube/config"
   openshift_pull_secret_path="~/pull-secret.txt"
   [new_workers]
   Worker-0.test.flex.com
   Worker-1.test.flex.com
   ```

4. Run the playbook in CSAH node to add the RHEL nodes to the OCP cluster. Ensure to specify the Ansible inventory file that was created in <path> code. For example, <path> = /etc/ansible/hosts.

   ```
   #cd /usr/share/ansible/openshift-ansible
   #ansible-playbook -i <path> playbooks/scaleup.yml
   ```

For detailed information, see Adding RHEL compute machines to an OpenShift Container Platform cluster.
4.3 Verifying cluster setup

Verify if the cluster is up and running using OpenShift Cluster web console.

To verify the nodes using **Command line**, follow these steps:

1. Log in to OCP cluster.

   ```
   oc login -u <user name>
   ```

2. Check cluster nodes status.

   ```
   #oc get nodes
   NAME            STATUS    ROLES     AGE       VERSION
   master-0.test.flex.com Ready master 21d v1.14.6+9fb2d5cf9 master-
   1.test.flex.com   Ready    master 21d v1.14.6+9fb2d5cf9 master-
   2.test.flex.com   Ready    master 21d v1.14.6+9fb2d5cf9 master-
   0.test.flex.com   Ready    worker 19d v1.14.6+6ac6aa4b0 worker-
   1.test.flex.com   Ready    worker 19d v1.14.6+6ac6aa4b0 master-
Deploying and routing application on OCP cluster

5.1 Deploying Images

1. Log in to OCP cluster and check the status.
   
   oc login -u <user name>

2. Verify cluster nodes status.

   #oc get nodes
   #oc new-project <project name>

   NAME                     STATUS   ROLES    AGE   VERSION
   0.test.flex.com   Ready    master   21d   v1.14.6+9fb2d5cf9
   1.test.flex.com   Ready    master   21d   v1.14.6+9fb2d5cf9
   2.test.flex.com   Ready    master   21d   v1.14.6+9fb2d5cf9
   0.test.flex.com   Ready    worker   19d   v1.14.6+6ac6aa4b0
   1.test.flex.com   Ready    worker   19d   v1.14.6+6ac6aa4b0

3. Create a project.

   Example: # oc new-project vxflex-test

4. Create an app.

   oc new-app <image-name> ex: # oc new-app openshift/hello
   # oc get pods

   NAME                     READY   STATUS      RESTARTS   AGE
   openshift-1-deploy   0/1     Completed   0          3m19s
   openshift-1-lqsg5    1/1     Running     0          3m11s

5.2 Deploy applications using S2I

Using Source-to-Image (S2I) tool to deploy your application from docker or git hub registries.

Example:

   #oc new-app https://github.com/openshift/ruby-hello-world -l name=hello-world
   #oc get pods ruby-hello-world-1-5vg5c   1/1     Running     0
   42s  ruby-hello-world-1-build   0/1     Completed   0
   2m25s  ruby-hello-world-1-deploy  0/1     Completed   0
   50s
5.3 Application routing

1. Check the service of the application.

```bash
# oc get svc
NAME             TYPE      CLUSTER-IP      EXTERNAL-IP   PORT(S)     AGE
ruby-hello-world  ClusterIP  172.30.68.161  <none>       8080/TCP     3m4s
```

2. Expose the route for service of the application.

```bash
# oc expose svc/ruby-hello-world
route.route.openshift.io/ruby-hello-world exposed
```

3. Check the routes that were created.

```bash
# oc get routes
NAME   HOST/PORT     PATH   SERVICES   PORT    TERMINATION   WILDCARD
ruby-hello-world ruby-hello-world-vxflex-test.apps.test.flex.com 8080-tcp  None

```

For more details about Deploying Applications, see Red Hat Application Deployment Guide.
Conclusion

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-premises container platform solution to meet their needs.
Appendix: Sample file

Here, a sample `install-config.yaml` file is added for reference purpose only.

```yaml
apiVersion: v1
baseDomain: flex.com
compute:
- hyperthreading: Enabled
  name: worker
  replicas: 0  # No. of Worker nodes.
controlPlane:
  hyperthreading: Enabled
  name: master
  replicas: 3  # No. of master nodes
metadata:
  name: test
networking:
  clusterNetworks:
  - cidr: 10.128.0.0/14
    hostPrefix: 23
    networkType: OpenShiftSDN
  serviceNetwork:
  - 172.30.0.0/16
platform:
  none: {}
```
Appendix: Technical support and resources

Dell.com/support is focused on meeting customer needs with proven services and support.

Storage technical documents and videos provide expertise that helps to ensure customer success on Dell EMC storage platforms.

B.1 Related resources

- Dell EMC VxFlex Ready Nodes
- VxFlex OS Documentation
- VxFlex Ready Node Documentation
- Dell EMC VxFlex
- Red Hat OpenShift 4.2 Documentation
- https://access.redhat.com/articles/4207611