Red Hat OpenShift Container Platform 4.12 on Oracle Cloud VMware Solution

VMware App Modernization

Table of contents

Red Hat OpenShift Container Platform 4.12 on Oracle Cloud VMware Solution

Executive Summary

Business Case

Red Hat® OpenShift® Container Platform offers automated installation, upgrades, and lifecycle management throughout the container stack—the operating system, Kubernetes and cluster services, and applications on any cloud. OpenShift helps teams build with speed, agility, confidence, and choice. OpenShift is focused on security at every level of the container stack and throughout the application lifecycle. It includes long-term and enterprise support from one of the leading Kubernetes contributors and open-source software companies.

The manageability of operating an OpenShift environment with virtualized infrastructure can be improved over the management of traditional IT infrastructure on bare metal, since the demand for resources can fluctuate with business needs, leaving the OpenShift cluster either under-powered or over-provisioned. IT needs a more flexible, scalable, and secure infrastructure to handle the ever-changing demands of OpenShift. With a single architecture that is easy to deploy, Oracle Cloud VMware Solution™ can provision compute, network, and storage on demand. Oracle Cloud VMware Solution protects network and data with micro-segmentation and satisfies compliance requirements with data-at-rest encryption. Policy-based management delivers business-critical performance.

In this solution, we provide the generic design and deployment guidelines for running Red Hat OpenShift on Oracle Cloud VMware Solution.

Why Run Red Hat OpenShift on Oracle Cloud VMware Solution?

Oracle Cloud VMware Solution combines automation with a standardized and repeatable approach to infrastructure, giving IT Operations the infrastructure agility necessary to support developers by providing developer-ready infrastructure.

The infrastructure automation capabilities of Oracle Cloud VMware Solution enable administrators to quickly deploy, manage, and scale the underlying infrastructure with cloud-like agility and at the speed of business.

By running OpenShift on Oracle Cloud VMware Solution, you get all the benefits of a modern public cloud based on the proven VMware Software-Defined Data Center architecture:

- A consistent and repeatable approach to standardized Infrastructure.
- Automation eliminates human error and fosters admin productivity.
- Cloud scale and agility enable easy to scale at the speed of the business.
- Integrate with the well-proven networking solution from VMware: VMware NSX™.
- Integrate with the well-proven storage solution from VMware: vSAN™.
- Leverage the advantages of VMware vSphere® features, such as VMware vSphere Distributed Resource Scheduler™ (DRS), vSphere vMotion®, vSphere HA, etc.

Audience

This reference architecture paper is intended for the following audiences:

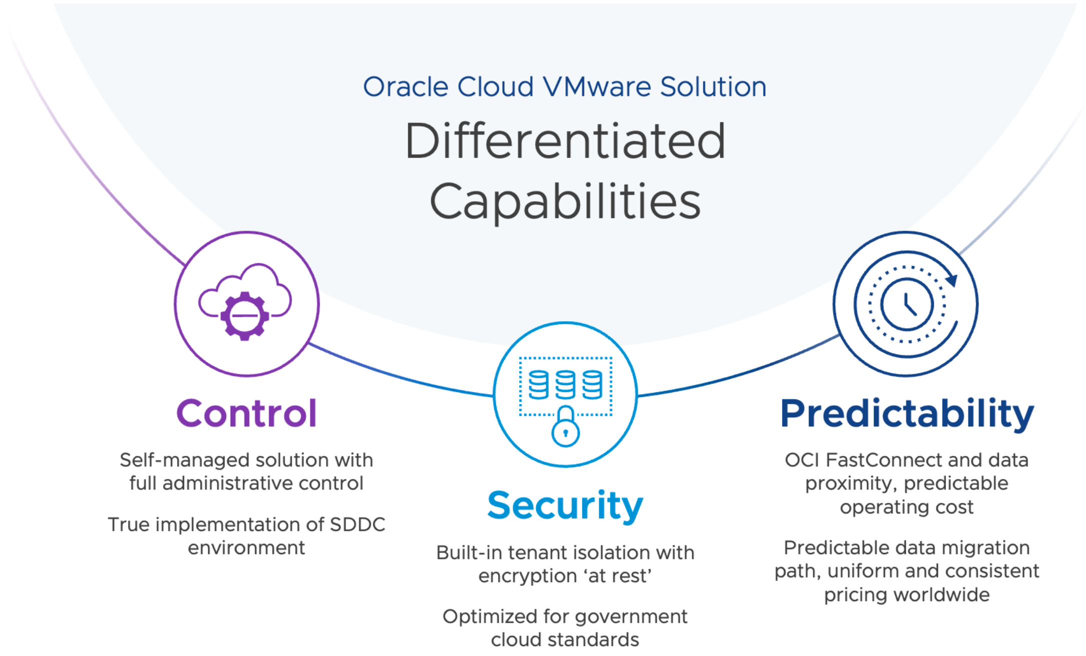
- Corporate CTOs and CIOs whose organizations run OpenShift or Kubernetes in a private datacentre but are exploring public cloud deployments.
- vSphere VI administrators who are familiar with VMware virtualized infrastructure and need to deploy and manage OpenShift in a virtualized environment.
- DevOps who are deploying, managing, or using OpenShift on vSphere.
- Any other engineer/operator/end-user who is interested in OpenShift/Kubernetes/vSphere and has basic knowledge about Oracle Cloud VMware Solution, vSAN, NSX, NSX Container Plug-in (NCP), Cloud Native Storage (CNS), Container Storage Interface (CSI), OpenShift, and Kubernetes.

Technology Overview

Solution technology components are listed below:

- Oracle Cloud VMware Solution
    - VMware vSphere
    - VMware vSAN
    - VMware NSX Data Center
- vSphere CSI for Kubernetes Storage
- Red Hat OpenShift Container Platform
- VMware NSX Container Plug-in for OpenShift

Oracle Cloud VMware Solution


Oracle Cloud VMware Solution
Differentiated Capabilities
Control
Self-managed solution with full administrative control
True implementation of SDDC environment
Security
Built-in tenant isolation with encryption 'at rest'
Optimized for government cloud standards
Predictability
OCI FastConnect and data proximity, predictable operating cost
Predictable data migration path, uniform and consistent pricing worldwide

Oracle Cloud VMware Solution allows you to create and manage VMware enabled software-defined data centers (SDDCs) in Oracle Cloud Infrastructure. Oracle Cloud VMware Solution (OCVS) enables customers to move VMware workloads to Oracle Cloud with minimal application re-architecture. Customers gain flexibility, scale, and agility while maintaining continuity from existing VMware based tools, processes, and policies. Customers have full administrative control of their VMware environments, while leveraging Oracle Cloud Native services to modernize applications.

VMware vSphere

VMware vSphere is VMware's virtualization platform, which transforms data centres into aggregated computing infrastructures that include CPU, storage, and networking resources. vSphere manages these infrastructures as a unified operating environment and provides operators with the tools to administer the data centres that participate in that environment. The two core components of vSphere are ESXi™ and vCenter Server®. ESXi is the hypervisor platform used to create and run virtualized workloads. vCenter Server is the management plane for the hosts and workloads running on the ESXi hosts.

VMware vSAN

VMware vSAN is the industry-leading software powering VMware's software defined storage and Hyperconverged Infrastructure (HCI) solution. vSAN helps customers evolve their data center without risk, control IT costs, and scale to tomorrow's business needs. vSAN, native to the market-leading hypervisor, delivers flash-optimized, secure storage for all of your critical vSphere workloads, and is built on industry-standard x86 servers and components that help lower TCO in comparison to traditional storage. It delivers the agility to scale IT easily and offers the industry's first native HCI encryption.

VMware NSX Data Center

VMware NSX Data Center is the network virtualization and security platform that enables the virtual cloud network, a software-defined approach to networking that extends across data centers, clouds, and application frameworks. With NSX Data Center, networking and security are brought closer to the application wherever it's running, from virtual machines to containers to bare metal. Like the operational model of VMs, networks can be provisioned and managed independently of the underlying hardware. NSX Data Center reproduces the entire network model in software, enabling any network topology—from simple to complex multitier networks—to be created and provisioned in seconds. Users can create multiple virtual networks with diverse requirements, leveraging a combination of the services offered via NSX or from a broad ecosystem of third-party integrations ranging from next-generation firewalls to performance management solutions to build inherently more agile and secure environments. These services can then be extended to a variety of endpoints within and across clouds.

Kubernetes vSphere CSI Driver

Cloud Native Storage (CNS) is a vSphere and Kubernetes (K8s) feature that makes K8s aware of how to provision storage on vSphere on-demand, in a fully automated, scalable fashion as well as providing visibility for the administrator into container volumes through the CNS User Interface within vCenter. Run, monitor, and manage containers and virtual machines on the same platform—in the same way:

- Simplify your infrastructure needs, lifecycle, and operations.
- Lower costs, using a platform you already know for consistent operations across workloads and across clouds.
- Spend less time managing infrastructure and more time building apps that provide business value.

The main goal of CNS is to make vSphere and vSphere storage, including vSAN, a platform to run stateful Kubernetes workloads. vSphere's data path is highly reliable, highly performant, and mature for enterprise. CNS enables access of this data path to Kubernetes and brings an understanding of Kubernetes volume and pod abstractions to vSphere.

See https://www.vmware.com/products/cloud-native-storage.html for detailed information regarding CNS.

Red Hat OpenShift Container Platform

Red Hat OpenShift Container Platform ships with Red Hat Enterprise Linux® CoreOS for the Kubernetes control plane nodes and supports both Red Hat Enterprise Linux CoreOS and Red Hat Enterprise Linux for worker nodes. OpenShift supports the Open Container Initiative (OCI), which is an open governance structure around container formats and runtimes. OpenShift includes hundreds of fixes to defect, security, and performance issues for upstream Kubernetes in every release. It is tested with dozens of technologies and is a robust tightly integrated platform. OpenShift includes software-defined networking and validates additional common networking solutions. OpenShift also validates numerous storage and third-party plug-ins for every release.

See https://www.openshift.com/products/container-platform for detailed information regarding OpenShift Container Platform.

VMware NSX Container Plug-in for OpenShift

VMware NSX Container Plugin (NCP) provides the integration between NSX Data Center and container orchestrators such as Kubernetes, as well as integration between NSX Data Center and container-based PaaS (platform as a service) software products such as OpenShift.

The main component of NCP runs in a container and communicates with NSX Manager and with the OpenShift control plane. NCP monitors changes to containers and other resources and manages networking resources such as logical ports, switches, routers, and security groups for the containers by calling the NSX Policy API.

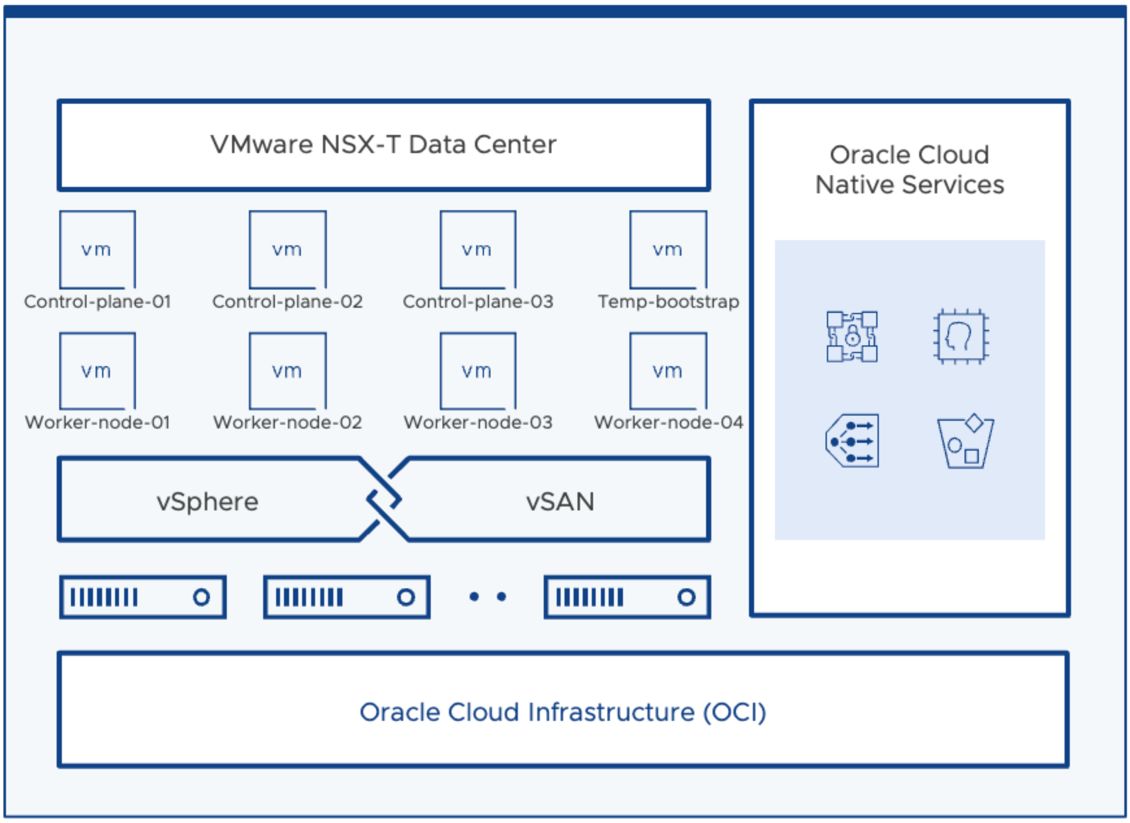
The NSX CNI plug-in runs on each OpenShift node. It monitors container life cycle events, connects a container interface to the guest vSwitch, and programs the guest vSwitch to tag and forward container traffic between the container interfaces and the vNIC.

See VMware Documentation for detailed information regarding NCP.

Architecture Diagram

In this solution, the Oracle Cloud VMware Solution test environment was composed of a single SDDC.

We deployed the OpenShift Container Platform into the SDDC, which contains the VMware NSX® Edge™ node alongside the other infrastructure VMs.

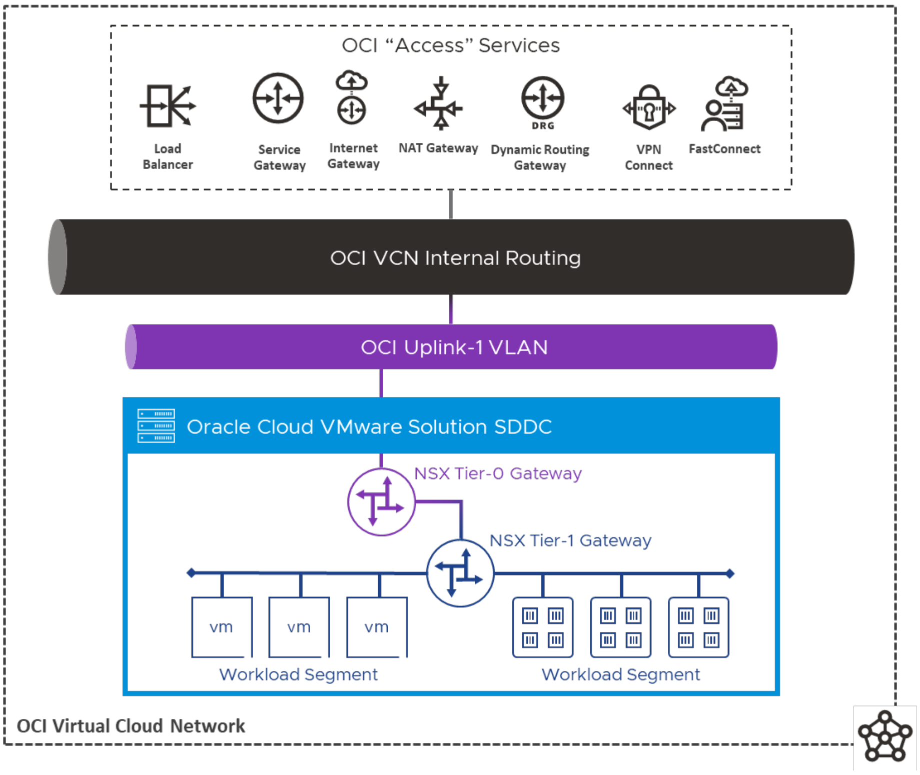


Notation in Figure 1:

- Temp Bootstrap: This is the temporary bootstrap node. It only exists during the installation process and will be deleted after the OpenShift Container Platform is fully deployed.
- Control plane-01,02,03: These are the control plane nodes of Kubernetes deployed and managed by OpenShift.
- Worker-node-01,02,03,04: These are the worker nodes of Kubernetes deployed and managed by OpenShift. We deployed 3 worker nodes as the starting point. More worker nodes can be added on demand through the OpenShift control plane.

Oracle Cloud VMware Solution Networking

The figure below shows the logical connectivity from workloads within the OCVS SDDC to the rest of OCI and beyond.



The Oracle Cloud VMware solution is a first-class, or "native" service so connects to all the other OCI services directly through the Virtual Cloud Network internal routing. It can also access the OCI connectivity or "access" services the same way.

OCVS Standard Bill of Material

VM Role	Minimum vCPU	Minimum Memory (GB)	Storage	VM Count
VMware vCenter	8	28	1TB	1
VMware NSX Manager Nodes	18	72	0.9TB	3
VMware NSX Edge Nodes	16	64	0.4TB	2
VMware HCX Manager	4	12	60GB	1
OpenShift Control Plane Nodes	4	16	120GB for OS	3
OpenShift Compute Nodes	2	8	120GB for OS	Minimum of 2 for a standard OpenShift cluster
OpenShift Bootstrap Node (Temporary)	4	16	120GB for OS	1

OCVS Installation

Oracle Cloud VMware Solution is deployed as a service within Oracle Cloud Infrastructure (OCI). Although outside the scope of this document setting up the OCI Virtual Cloud Network (VCN) within an Oracle Cloud "Tenancy" is described on the Networking section of the Oracle Documentation site. The VCN contains the network address space from which the OCVS infrastructure deployment is assigned. The address space used by workloads within OCVS must be separate from the parent VCN addressing.

The provisioning of an OCVS SDDC can be accomplished from the Oracle Cloud Console, or, programmatically using OCI-CLI, the provided SDK bindings or using the OCI Terraform Provider. Whichever route is taken, the process is broadly similar with the number of hosts, their "shape" (Oracle Cloud Compute server specification) and network details provided to the Oracle Cloud Provisioning Service which then builds the SDDC accordingly. This process is described in the Overview of Oracle Cloud VMware Solution section of the Oracle Documentation site.

Network Configurations

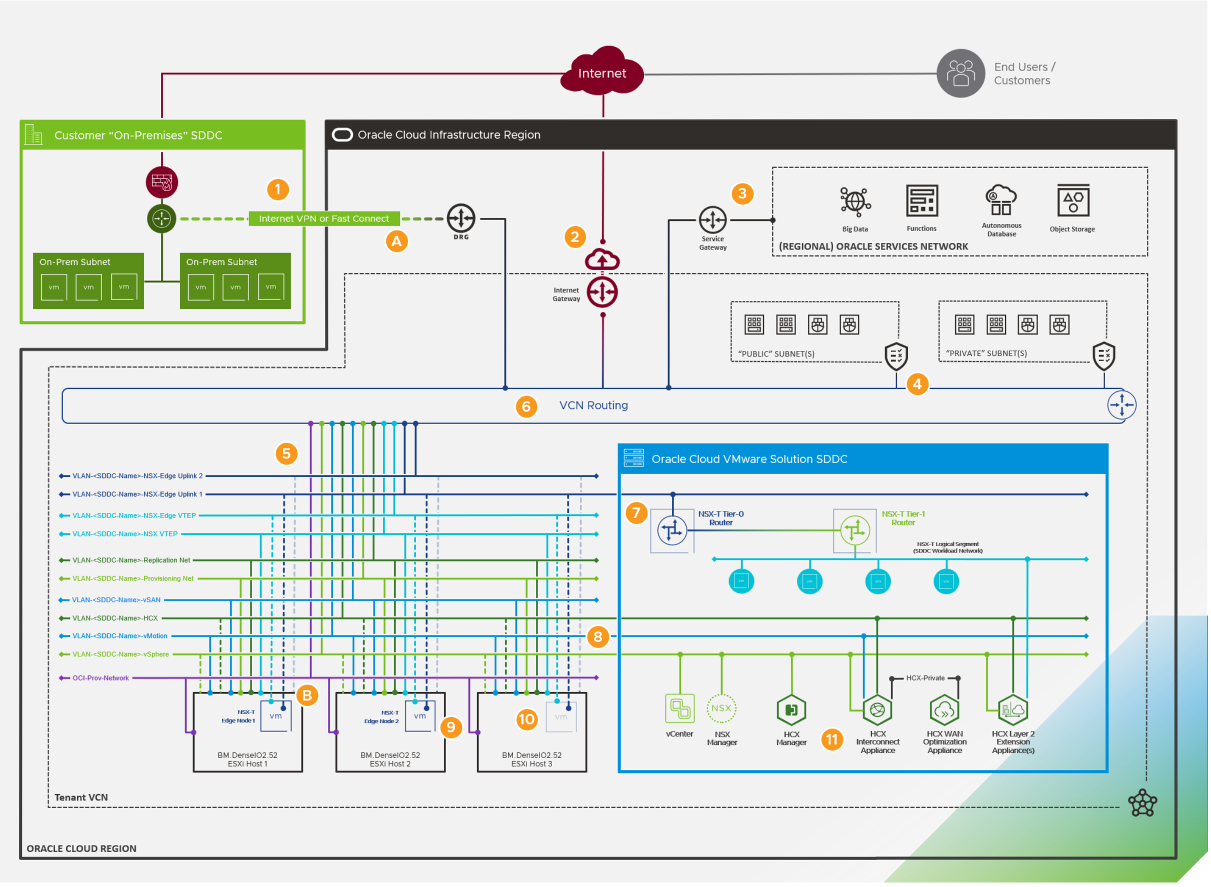
Oracle Cloud VMware Solution is connected to the outside world through the OCI VCN within which it is provisioned. Native routing to and from the SDDC is available from the local VCN, connected VCNs within the same Oracle Cloud Region, or, other remote regions, on-premises networks or datacentres. Public Internet access can be provided locally with OCI or, if preferred, via an existing corporate Internet Gateway.

Outbound Internet access is available within OCI using the "NAT Gateway" service to provide access to workloads which do not require inbound public connectivity and can share a single, ephemeral, public Internet address. Inbound (and Outbound) Internet access can be provided using OCI's "Internet Gateway" service where each accessible service or workload is assigned a dedicated public Internet address.

Routing to and from the local VCN and other "on-net" destinations can be accomplished using compatible corporate network addresses assigned to services which route without the need for NAT to and from any connected networks. On-premises networks can be connected to Oracle Cloud using the dedicated connectivity "FastConnect" service or, over the Internet using OCI's IPSec-based "VPN Connect" service.

Network services can be provided by OCI's range of capabilities or, within the SDDC using the capabilities inherent by the provided NSX environment. Additional services can be deployed into the OCI or OCVS environments in the form of virtual appliances in cases where specialized capabilities are required.

The graphic below shows an overview of the networking layout and connectivity within OCI and OCVS. You can see the full original image in this VMware blog post.


Internet
End Users / Customers
Customer "On-Premises" SDDC
1
Internet VPN or Fast Connect
A
On-Prem Subnet
vm
vm
vm
On-Prem Subnet
vm
vm
vm
Oracle Cloud Infrastructure Region
DRG
2
Internet Gateway
3
Service Gateway
Big Data
Functions
Autonomous Database
Object Storage
(REGIONAL) ORACLE SERVICES NETWORK
"PUBLIC" SUBNET(S)
"PRIVATE" SUBNET(S)
4
6
VCN Routing
5
VLAN-<SDDC-Name>-NSX-Edge Uplink 2
VLAN-<SDDC-Name>-NSX-Edge Uplink 1
VLAN-<SDDC-Name>-NSX-Edge VTEP
VLAN-<SDDC-Name>-NSX VTEP
VLAN-<SDDC-Name>-Replication Net
VLAN-<SDDC-Name>-Provisioning Net
VLAN-<SDDC-Name>-vSAN
VLAN-<SDDC-Name>-HCX
VLAN-<SDDC-Name>-vMotion
VLAN-<SDDC-Name>-vSphere
OCI-Prov-Network
B
NSX-T Edge Node 1
vm
BM.DenseIO2.52 ESXi Host 1
NSX-T Edge Node 2
vm
9
BM.DenseIO2.52 ESXi Host 2
10
vm
BM.DenseIO2.52 ESXi Host 3
Oracle Cloud VMware Solution SDDC
7
NSX-T Tier-0 Router
NSX-T Tier-1 Router
NSX-T Logical Segment (SDDC Workload Network)
vm
vm
vm
vm
8
HCX-Private
vCenter
NSX
NSX Manager
HCX Manager
11
HCX Interconnect Appliance
HCX WAN Optimization Appliance
HCX Layer 2 Extension Appliance(s)
Tenant VCN
ORACLE CLOUD REGION

If you wish to consume the NSX Container plugin, please reference the NSX Container Plugin for OpenShift -Installation and Administration Guide to create necessary NSX components, such as tier-0 and tier-1 routers.

Storage Configuration

With full “root” access to the OCVS SDDC it is possible to configure the vSAN cluster in any supported configuration. At deployment time the SDDC’s vSAN uses the “default” storage policy detailed in the validation test specifications below, but this can be amended by the customer to suit local preferences.

The validation tests were conducted using the default vSAN datastore storage policy of RAID 1 FTT=1, checksums enabled. The vSAN cluster has deduplication and compression deactivated, and no encryption. In the below sections, we explained the detailed configurations of the vSAN cluster and some items in the Storage Policy Based Management (SPBM).

Deduplication and Compression

The ‘Deduplication and Compression’ option was configured on the cluster level and it can be enabled or deactivated for the whole vSAN cluster. While in our testing we deactivated it, by enabling it we can reduce the vSAN storage usage but induce higher latencies for the OpenShift application. This is a tradeoff for customers’ choices.

Failures to Tolerate (FTT)

Failures to Tolerate (FTT) is a configuration item in vSAN’s storage policy. For the ‘StorageClass’ in OpenShift and the corresponding vSAN’s storage policy, we recommended setting vSAN’s Failures to Tolerate (FTT) to 1. In our testing, we set FTT to 1 as the baseline. Do not set the FTT to 0 in an OpenShift with vSAN deployment because FTT=0 may possibly cause the data of the replications of the same pod to be stored in the same physical disk. This may cause data loss in case of a physical disk failure.

In the case of using RAID 1 in vSAN policy, there are two copies for each piece of data in vSAN. So, the estimated database capacity requirement should not exceed half of the vSAN’s overall capacity. In the case of RAID 5, vSAN consumes 1.33 times of the raw capacity and you can calculate the storage usage accordingly. If the capacity increase is needed, the additional machines can be added to the cluster and vSAN can increase the data capacity storage for OpenShift online without the service interruption to OpenShift users.

Erasure Coding (RAID 1 vs. RAID 5)

Erasure Coding is a configuration item in vSAN’s storage policy. It is also known as configuring RAID 5 or RAID 6 for vSAN objects. With FTT=1 and RAID 1, the data in vSAN is mirrored and the capacity cost would be 2 times of the raw capacity. With FTT=1 and RAID 5, the data is stored as RAID 5 and the capacity cost would be 1.33 times of the raw capacity.

In our testing, we used FTT=1 without Erasure Coding (RAID 1). By enabling Erasure Coding, we could save some vSAN storage spaces but induce higher latencies for the Kubernetes applications. Again, this is a trade-off for customers’ choices.

Openshift Installation

Pre-requisites

The following pre-requisites were put into place before deploying the Red Hat OpenShift Cluster.

Obtain a copy of a valid Red Hat OpenShift Pull Secret file from the Red Hat Cloud Console. This file is used to authenticate the cluster with Red Hat services for deployment and updates. Red Hat does separately provide an air-gap installation method for new clusters where internet connectivity is not available, please refer to their documentation.

A new NSX segment was created, named “192.168.200.0/24” with a gateway and network address of 192.168.200.1/24. This network will be used to deploy the OpenShift Cluster nodes to.

A Linux server was deployed to this new network segment with the following roles enabled; DNS, DHCP.

The following Host A DNS records were created, pointing to a reserved IP address on the same network subnet which the OpenShift nodes are deployed to:

- api.{cluster_name}.{base_domain}
  - Example – api.ocp-ocvs.veducate.local – 192.168.200.40
- *.apps.{cluster_name}.{base_domain}
  - Example - *.apps.ocp-ocvs.veducate.local

The DHCP service was configured with the following address pool, 192.168.200.100-192.168.200.200.

NSX Distributed firewall rules were created to identify the subnet group 192.168.200.0/24 and allow outbound HTTPs access to the public internet. Another firewall rule was created to allow DNS requests from the Linux Server to an upstream public DNS service.

- Please note that in a production environment, it is best practice to implement stricter micro-segmention rules within your subnets and to any traffic that leaves the internal network.

Finally, a second Linux Server was deployed and configured with the following, to be purposed as a bastion host into the environment.

- OpenShift-Install CLI tool
- OpenShift Client CLI tool
- Kubectl CLI tool
- SSH Server configured
- Download and install the certificates from your vCenter endpoint
  - curl -O https://{vCenter_FQDN}/certs/download.zip
  - Follow your chosen operating systems guide on installing the certificates to the trusted store

Installing Red Hat OpenShift using the Openshift-Install Tool

In this validation, a “Installer provisioned Installation” method was followed, which deploys a reference architecture deployment of Red Hat OpenShift against your chosen platform, with the necessary infrastructure components as part of that installation.

When performing this type of installation, there are two ways in which to consume the “openshift-install” tool. The first is to run the wizard, which will provide several queries such as domain name, cluster name, vCenter details and authentication and pullSecret file. The second, is to create a YAML file as described below. This allows for more flexibility and control, such as specifying the VM resources for the cluster nodes. If you wish to integrate with NSX, then you will need this later method, which can be initated by either running the command “openshift-install create install-config” which will output an “install-config.yaml” file for you to edit or by directly creating the form yourself using the same name.

```
apiVersion: v1
baseDomain: veducate.local
compute:
- hyperthreading: Enabled
  name: worker
  replicas: 1
  platform:
    vsphere:
      cpus: 8
      coresPerSocket: 4
      memoryMB: 16384
      osDisk:
        diskSizeGB: 120
controlPlane:
  hyperthreading: Enabled
  name: master
  replicas: 3
  platform:
    vsphere:
      cpus: 8
      coresPerSocket: 4
      memoryMB: 16384
      osDisk:
        diskSizeGB: 120
metadata:
  creationTimestamp: null
  name: ocp-ocvs
networking:
  clusterNetwork:
  - cidr: 10.128.0.0/14
    hostPrefix: 23
  machineNetwork:
  - cidr: 10.0.0.0/16
  networkType: OpenShiftSDN
  serviceNetwork:
  - 172.30.0.0/16
platform:
  vsphere:
    apiVIP: 192.168.200.192
    cluster: Cluster-1
    folder: /vEducate-DC/vm/OpenShift/
    datacenter: vEducate-DC
    defaultDatastore: Datastore01
    ingressVIP: 192.168.200.193
    network: "network_NW1"
    password: Password@!
    username: admin@veducate.local
    vCenter: vcenter.veducate.local
publish: External
pullSecret:
'{"auths":{"cloud.openshift.com":{"auth":"bxxxxxx==","email":"openshift@veducate.co.uk"},"registry.redhat.io":{"auth":
"Nxxx=","email":"openshift@veducate.co.uk"}}}'
sshKey: |
  ssh-rsa AAAABxxxxxx openshift@veducate
```

Both the installation wizard, or consuming an existing “install-config.yaml” file, can be achieved by running “openshift-install create cluster”.

The installation tool will provide console output, such as the below example, however this level of detail can be changed by using the command line argument “—log-level={level}”.

```
INFO Consuming Install Config from target directory

INFO Creating infrastructure resources...

INFO Waiting up to 20m0s (until 9:52AM) for the Kubernetes API at https://api.ocp-ocvs.veducate.local:6443...

INFO API v1.25.4+18eadca up

INFO Waiting up to 30m0s (until 10:04AM) for bootstrapping to complete...

INFO Destroying the bootstrap resources...

INFO Waiting up to 40m0s (until 10:30AM) for the cluster at https://api.ocp-ocvs.veducate.local:6443 to initialize...

INFO Checking to see if there is a route at openshift-console/console...

INFO Install complete!

INFO To access the cluster as the system:admin user when using 'oc', run 'export KUBECONFIG=/home/ocp-
ocvs/auth/kubeconfig'
```

```
INFO Access the OpenShift web-console here: https://console-openshift-console.apps.ocp-ocvs.veducate.local

INFO Login to the console with user: "kubeadmin", and password: "ur6xT-gxmVW-WVUuD-Sd44J"

INFO Time elapsed: 35m16s
```

At this point, you can now interact with the cluster, as per the output, you can either use the provided kubeconfig file, or the OpenShift Console UI.

This installation has been validated jointly by Red Hat and VMware. While this is a fully supported deployment of Red Hat OpenShift on vSphere provided by OVCS, we wanted to dedicate extra time to validate that the OpenShift deployment runs as expected in the OCVS infrastructure and that it successfully passes the conformance tests used internally at Red Hat.

Please note that this applies to any installation method you decide to use to deploy OpenShift on vSphere. For detailed instructions please refer to Preparing to install on vSphere.

References

- VMware vSphere
- VMware vSAN
- VMware NSX Data Center
- Red Hat OpenShift Container Platform

Authors and Contributors

- Dean Lewis, Senior Specialist Solution Engineer – Multi-Cloud Management, VMware
- Steve Dockar, Worldwide OCVS Field Solution Architect Director, VMware
- Ramon Acedo Rodriguez – Senior Principal Product Manager, Red Hat
- Vivien Wang – Engineering Partner Manager, Red Hat
- Jatin Purohit, Sr. Technical Marketing Architect, VMware