The science behind the report:

VMware vSphere 7 Update 2 offered greater VM density and increased availability compared to OpenShift Virtualization on Red Hat OpenShift 4.9

This document describes what we tested, how we tested, and what we found. To learn how these facts translate into real-world benefits, read the report VMware vSphere 7 Update 2 offered greater VM density and increased availability compared to OpenShift Virtualization on Red Hat OpenShift 4.9.

We concluded our hands-on testing on January 31, 2022. During testing, we determined the appropriate hardware and software configurations and applied updates as they became available. The results in this report reflect configurations that we finalized on October 15, 2021 or earlier. Unavoidably, these configurations may not represent the latest versions available when this report appears.

Our results

To learn more about how we have calculated the wins in this report, go to http://facts.pt/calculating-and-highlighting-wins. Unless we state otherwise, we have followed the rules and principles we outline in that document.

Table 1: Results of our testing

	VMware®vSphere® 7 Update 2	Red Hat® OpenShift® 4.9
Resource management tests		
Maximum number of active SQL Server VM and client VM pairs each solution supported	30	14
Maximum number of idle VMs each solution supported	245	61
Hands-on admin time required to balance load across VMs	N/A	02:58
Management tasks		
High availability (HA) functionality		
Time to configure HA for the cluster	00:26	00:34
Time for a VM to fail over and restart	01:42	06:34
Live migration functionality		
Time to migrate VM from one host to another	01:17	01:57
Downtime during storage live migration	N/A	10:17
VM functionality		
Downtime while expanding VM virtual hardware	N/A	00:59
Downtime while creating a snapshot of a VM	N/A	01:11
Time to enable VM encryption	01:02	06:04

System configuration information

Table 2: Detailed information on the system we tested.

System configuration information	HPE ProLiant DL380 Gen10
BIOS name and version	U30 v2.42
Operating system name and version/build number	VMware ESXi™ 7.0.2 Red Hat Enterprise Linux® CoreOS 49.84.202110081407-0
Date of last OS updates/patches applied	10/8/2021
Power management policy	Static high performance mode
Processor	
Number of processors	2
Vendor and model	Intel® Xeon® Gold 5120
Core count (per processor)	14
Core frequency (GHz)	2.2
Stepping	M0
Memory module(s)	
Total memory in system (GB)	256
Number of memory modules	4
Vendor and model	Hynix® 809085-091
Size (GB)	64
Type	PC4-19200T
Speed (MHz)	2,400
Speed running in the server (MHz)	2.400
Storage controller	
Vendor and model	HPE Smart Array P408i-a SR Gen10
Cache size (GB)	N/A
Firmware version	3.53
Local storage (capacity)	
Number of drives	2
Drive vendor and model	Intel S4500 SSDSC2KG96
Drive size (GB)	960
Drive information (speed, interface, type)	6Gbps, SATA, SSD

System configuration information	HPE ProLiant DL380 Gen10
Local storage (cache)	
Number of drives	2
Drive vendor and model	Intel S3700 SSDSC2BA40
Drive size (GB)	400
Drive information (speed, interface, type)	6Gbps, SATA, SSD
Network adapter 1	
Vendor and model	HPE 331i
Number and type of ports	4x 1Gb Ethernet
Driver version	20.18.31
Network adapter 2	
Vendor and model	Mellanox® ConnectX-3 Ethernet Adapter
Number and type of ports	2x 10GbE SFP
Driver version	02.40.70.04
Cooling fans	
Vendor and model	HPE PFR0612XHE
Number of cooling fans	6
Power supplies	
Vendor and model	HPE 865408-B21
Number of power supplies	2
Wattage of each (W)	500

How we tested

Setting up and preparing the VMware vSphere 7 Update 2 (7U2) environment

Installing ESXi

1. Boot into the ESXi install ISO.
2. In the installation welcome screen, to continue, press Enter.
3. In the EULA, to accept and continue press F11.
4. Select the correct drive. To continue, press Enter.
5. Select the keyboard layout. To continue, press Enter.
6. Enter a root password. To continue, press Enter.
7. In the confirm install screen, to install press F11.
8. For the remaining three hosts, complete steps 1 through 7.

Configuring ESXi

1. Log into the ESXi server console.
2. Select Configure Management Network.
3. Select IPv4 Configuration.
4. In IPv4 Configuration, select a static IPv4 address, enter the appropriate IP address, and press Enter.
5. Select DNS Configuration.
6. In DNS Configuration, enter your Primary DNS server and fully qualified domain name, and press Enter.
7. To exit the management network configuration, press Esc.
8. To apply changes and restart the management network, press Y when prompted.
9. Select Troubleshooting Options.
10. In Troubleshooting Mode Options, enable ESXi Shell and SSH, and press Esc to confirm.
11. For the remaining four hosts, complete steps 1 through 10.

Enabling Network Time Protocol (NTP) on ESXi

1. Log into the ESXi server GUI.
2. Click Manage.
3. Click Time & date.
4. Click Edit NTP Settings.
5. In Edit time configuration, select Use Network Time Protocol (enable NTP client), select Start and stop with host, add your NTP server, and click Save.
6. For the remaining four hosts, complete steps 1 through 5.

Installing the vCenter Server Appliance

1. Download VMware vCenter® from the VMware support portal at https://my.vmware.com.
2. Mount the image on your local system, and browse to the vcsa-ui-installer folder. Expand the folder for your OS, and launch the installer if it doesn't automatically begin.
3. Once the vCenter Server Installer wizard opens, click Install.
4. To begin installation of the new vCenter Server Appliance™, click Next.
5. To accept the license agreement check the box, and click Next.
6. Enter the IP address of a temporary ESXi host. Provide the root password, and click Next.
7. To accept the SHA1 thumbprint of the server's certificate, click Yes.
8. Accept the VM name, and provide and confirm the root password for the VCSA. Click Next.
9. Set the size for the environment you're planning to deploy, check Thin Disk, and click Next.
10. Select the datastore on which to install. Accept the datastore defaults, and click Next.
11. Enter the FQDN, IP address information, and DNS servers you want to use for the vCenter Server Appliance. Click Next.
12. To begin deployment, click Finish.
13. When Stage 1 completes, click Close. To confirm, click Yes.
14. Open a browser window, and navigate to `https://[vcenter.FQDN]:5480/`.
15. On the Getting Started - vCenter Server page, click Set up.
16. Enter the root password, and click Log in.
17. Click Next.

18. Enable SSH access, and click Next.
19. To confirm the changes, click OK.
20. For the Single Sign-On domain name, type `vsphere.local`. Enter a password for the administrator account, confirm it, and click Next.
21. Click Next.
22. Click Finish.

Creating a cluster in vSphere

1. Open a browser, and enter the address of the vCenter server you deployed (`https://[vcenter.FQDN]/ui`).
2. In the left panel, select the vCenter server, right-click, and select New Datacenter.
3. Enter a name for the new data center, and click OK.
4. Select the data center you just created, right-click, and select New Cluster.
5. Enter a name for the cluster, and enable vSphere DRS, HA, and vSAN. Click OK.
6. In the cluster configuration panel, under Add hosts, click Add.
7. Check the box for Use the same credentials for all hosts. Enter the IP Address and root credentials for the first host, and the IP addresses of all remaining hosts. Click Next.
8. Check the box beside Hostname/IP Address to select all hosts, and click OK.
9. Click Next.
10. Click Finish.

Configuring a cluster via quickstart

1. Select Menu→ Hosts and Clusters.
2. Click the TKG Cluster.
3. In the right pane, the Quickstart view should automatically appear. Click Configure.
4. In Distributed switches, accept the defaults, add your host adapters, and click Next.
5. In vMotion traffic, complete the fields as follows, and click Next:
   - Use VLAN: `1612`
   - Protocol: IPv4
   - IP Type: Static
   - Host #1 IP: `192.168.12.11`
   - Host #1 subnet: `255.255.255.0`
   - Enable autofill
6. In Storage traffic, complete the fields as follows:
   - Use VLAN: `1613`
   - Protocol: IPv4
   - IP Type: Static
   - Host #1 IP: `192.168.13.11`
   - Host #1 subnet: `255.255.255.0`
   - Enable autofill
7. In Advanced options, leave defaults, and click Next.
8. In Claim disks, ensure the correct disks are selected, and click Next.
9. In Create fault domains, accept the defaults, and click Next.
10. To create the cluster, in Ready to complete, click Finish.

Creating and configuring VMs for the two density tests

Creating a client VM

1. Select Menu→ VMs and Templates.
2. Right-click the data center, and select New Virtual Machine.
3. Select Create a new virtual machine, and click Next.
4. Name your client VM, and click Next.
5. Choose your data center, and click Next.
6. Choose your vSAN storage, and click Next.
7. Leave default compatibility, and click Next.
8. For your OS, choose Windows 10, and click Next.

9. Make the following changes to the hardware:
    - CPU: 2
    - Memory: 8 GB
    - OS disk: 30 GB
10. Click Next.
11. To create the VM, click Finish.
12. Right-click the VM, and select Power On.
13. Right-click the VM, and open its console.
14. Mount the installation ISO to the VM, and restart it.
15. When prompted to boot from DVD, press any key.
16. When the installation screen appears, leave language, time/currency format, and input method as default, and click Next.
17. Click Install now.
18. When the installation prompts you, enter the product key.
19. Select Windows 10 Professional, and click Next.
20. Check I accept the license terms, and click Next.
21. Click Custom: Install Windows only (advanced).
22. Select Drive 0 Unallocated Space, and click Next. Windows will begin and restart automatically.
23. When the Settings page appears, enter the same password in the Password and Reenter Password fields.
24. Log in using the password you set up in the previous step.
25. Using Windows update, download and install all updates.
26. To ensure the system does not apply new updates during testing, disable the Windows Update service.

Installing HammerDB 4.2

1. On the client VM, download HammerDB from https://hammerdb.com/download.html.
2. Double-click the .exe file, choose English, and click OK.
3. Click Yes.
4. Click Next.
5. Chose the default destination location, and click Next.
6. Click Next.
7. Click Finish.

Creating a database VM

1. Select Menu→VMs and Templates.
2. Right-click the data center, and select New Virtual Machine.
3. Select Create a new virtual machine, and click Next.
4. Name your client VM, and click Next.
5. Choose your data center, and click Next.
6. Choose your vSAN storage, and click Next.
7. Leave default compatibility, and click Next.
8. For your OS, choose Windows Server 2019, and click Next.
9. Make the following changes to the hardware:
    - CPU: 8
    - Memory: 64 GB
    - OS disk: 50 GB
    - Database disk: 100 GB
    - Log disk: 50 GB
10. Click Next.
11. To create the VM, click Finish.
12. Right-click the VM, and select Power On.
13. Right-click the VM, and open the console to it.
14. Mount the installation ISO to the VM, and restart it.
15. When prompted to boot from DVD, press any key.
16. When the installation screen appears, leave language, time/currency format, and input method as default, and click Next.
17. Click Install now.

18. When the installation prompts you, enter the product key.
19. Select Windows Server 2019 Standard Edition (Desktop Experience), and click Next.
20. Check I accept the license terms, and click Next.
21. Click Custom: Install Windows only (advanced).
22. Select Drive 0 Unallocated Space, and click Next. Windows will begin and restart automatically.
23. When the Settings page appears, enter the same password in the Password and Reenter Password fields.
24. Log in with the password you set up in the previous step.
25. Right-click the Start Menu, and select Disk Management.
26. In Disk Management, right click the database drive, and select Online.
27. Right-click the database drive, and choose New Simple Volume.
28. In the New Volume screen, accept defaults, and click OK.
29. For the log volume, complete steps 25 through 28.

Configuring Microsoft Windows Server 2019

1. Open Server Manager, and click Local Server.
2. Disable IE Enhanced Security Configuration.
3. Change the time zone to your local time zone.
4. Change the name of your server. When prompted, reboot.
5. Open Server Manager again, and click Local Server.
6. To run updates, click the link.
7. Run updates, rebooting when prompted, until the server shows no new updates to install.
8. To ensure the system does not apply new updates during testing, disable the Windows Update service.

Installing Microsoft SQL Server 2019 Enterprise

1. Download or copy the ISO to the server, and unzip it.
2. Double-click the Setup application.
3. Click Installation→New SQL Server Standalone installation or add features to an existing installation.
4. Choose the trial version, and click Next.
5. Select I accept the license terms and Privacy Statement, and click Next.
6. Select Use Microsoft Update to check for updates (recommended), and click Next.
7. On the Install Rules page, click Next.
8. Select the following features, and click Next:
    - Database Engine Services
    - Full-Test and Semantic Extractions for Search
    - Client Tools Connectivity
    - Client Tools Backwards Compatibility
9. Leave the Default instance, and click Next.
10. Leave the default Service Accounts, and click Next.
11. On the Server Configuration tab, choose Mixed Mode, and enter and confirm a Password for the SQL Server system administrator (sa) account.
12. To specify the SQL Server administrators, click Add Current User.
13. Click Next.
14. Once you've passed the rule check, click Next.
15. Click Install.
16. Once the installation finishes, go back to the SQL Server Installation Center, and click Install SQL Server Management Tools.
17. Download the SSMS file, and install with defaults.
18. When prompted, reboot the server.
19. Run Windows Update one more time to ensure there aren't any new updates for SQL (make sure Windows Updates are set to get updates for other Microsoft products).
20. Once you've installed all available updates, disable Windows Update service by performing the following actions:
    - Click Start.
    - To open the Services list, type `services`.
    - Disable the Windows Update service.

Locking pages in memory

1. Click Start, and type `Local Security Policy`.
2. When the program pops up in the search, open it.
3. Expand Local Policies, and click User Rights Assignment.
4. In the right-hand pane, scroll down, and double-click Lock pages in memory.
5. Click Add User or Group, type `NT Service\MSSQLSERVER`, and click OK.
6. To close the Properties window, click OK.
7. Close the Local Security Policy window.

Configuring SQL Server on the VMs under test

1. Open the SQL Server Management Studio.
2. Right-click Databases, and select New Database.
3. In the New Database screen, name your database, and click Filegroups.
4. In Filegroups, on the database drive, create eight database files that are 8 GB each on the database drive. On the log drive, create one log file that is 15 GB.
5. Click OK.

Populating the database

1. Open HammerDB on the client system, and click Options→Benchmark.
2. Choose Microsoft SQL and TPC-C.
3. Expand Schema Build, and double-click Options.
4. Set the IP of your SQL Server.
5. Choose SQL User (SA) connection, and enter the password.
6. Set the number of warehouses to `500`.
7. Set the number of virtual users to `16`.
8. Click OK.
9. Expand Virtual Users, and double-click Create.
10. To start the build, click Run Virtual Users.

Backing up the database

1. On the SQL server, Open SQL Server Management Studio.
2. Right-click the TPC-C database, and click Tasks→Back up….
3. Select the database volume to store the backup file, enable compression, and click OK.

Performing the test

1. On the client system, start HammerDB.
2. Set the database server to SQL Server and the workload to TPC-C.
3. Open the Options panel for the Virtual Users: SQL Server→TPC-C→Virtual User→Options.
4. For virtual users type `16`.
5. Select the following:
   - Show Output
   - Log Output to Temp
   - Use Unique Log Name
   - Log Timestamps
6. Click OK.
7. Open the Options panel for the Driver: SQL Server→TPC-C→Driver Script→Options.
8. Enter the IP of the target SQL server.
9. Choose SQL Server Authentication connection, and enter the password.
10. Select Timed Driver Script, and type `15` for Rampup and `30` for the Test Duration.
11. Click Create Virtual Users.
12. To start the HammerDB workload, back on the client, click the green arrow.
13. When the run finishes, stop the data collector set on the target SQL Server.
14. Save the HammerDB results text file and data collector output for review.
15. Open SQL Manager, delete the TPCC database, and restore the database from the SQL backup file.
16. Reboot the SQL Server VM.

Setting up and preparing the Red Hat OpenShift environment

Deploying the testbed

Configuring DNS

1. Log into your domain controller.
2. Press the Start key, type `DNS`, and press Return.
3. In DNS manager, navigate to your domain's forward lookup zone.
4. Right-click your forward lookup zone, and select New Host (A or AAAA).
5. In the New Host screen, make sure that Create associated pointer (PTR) record is selected, enter the host name and IP address, and click Add Host.
6. To acknowledge the creation of the host, click OK.
7. Complete steps 5 and 6 for the following hosts:

api.openshift	10.209.60.201
api-int.openshift	10.209.60.201
*.apps.openshift	10.209.60.202
bootstrap.openshift	10.209.60.10
etcd-0.openshift	10.209.60.11
etcd-1.openshift	10.209.60.12
etcd-2.openshift	10.209.60.13
master-0.openshift	10.209.60.11
master-1.openshift	10.209.60.12
master-2.openshift	10.209.60.13
worker-0.openshift	10.209.60.21
worker-1.openshift	10.209.60.22
worker-2.openshift	10.209.60.23
worker-3.openshift	10.209.60.24
worker-4.openshift	10.209.60.25

Installing the Red Hat OpenShift provisioner

We installed our OpenShift installer on a physical host with a network connection to the infrastructure network.

Installing Red Hat Enterprise Linux

1. Connect a Red Hat Enterprise Linux 8.4 installation disk image to the host.
2. Boot the host off the Red Hat Enterprise Linux installation disk.
3. When choosing what to boot, select Install Red Hat Enterprise Linux 8.4.0.
4. To move past the start screen once the installer loads, click Next.
5. In the Installation Summary screen, click Installation Destination.
6. In the Installation Destination screen, accept the default, and click Done.
7. Back in the Installation Summary screen, click KDUMP.
8. In the KDUMP screen, uncheck Enable kdump, and click Done.
9. Back in the Installation Summary screen, click Software Selection.
10. Select Minimal Install, and click Done.
11. Back in the Installation Summary screen, click Network & Host Name.

12. In Network & Host Name, choose your hostname, turn your Ethernet connection on, and click Configure.
13. In Editing [your Ethernet connection], click Connect automatically with priority: 0, and click Save.
14. Click Done.
15. Back in the Installation Summary screen, click Begin Installation.
16. In the Configuration screen, click Root Password.
17. In Root Password, enter your root password, and click Done.
18. To reboot into the OS once the installation completes, click Reboot.

Creating an SSH private key

1. Log into the OpenShift installer.
2. Create a non-root user account with root access for OpenShift:

```
useradd kni
passwd kni
echo "kni ALL=(root) NOPASSWD:ALL" | tee -a /etc/sudoers.d/kni
chmod 0440 /etc/sudoers.d/kni
```

3. Create an SSH key for your new user that you will use for communication between your cluster hosts:

```
su - kni -c "ssh-keygen -t ed25519 -f /home/kni/.ssh/id_rsa -N ''"
```

When configuring your OpenShift install_config.yaml file, you will need to use the information in /home/kni/.ssh/id_rsa.pub to fill in the ssh key value.

Installing the OpenShift installer

1. On the OpenShift provisioner, log into the kni user.
2. Disable the firewall service:

```
sudo systemctl stop firewalld
sudo systemctl diable firealld
```

3. To allow your system to install updates, register your Red Hat Enterprise Linux installation to Red Hat.
4. Install the prerequisites for the OpenShift installation:

```
sudo dnf install -y libvirt qemu-kvm mkisofs python3-devel jq ipmitool
```

5. Modify and start the libvirtd service:

```
sudo usermod --append --groups libvirt kni
sudo systemctl enable libvirtd --now
```

6. Create the default storage pool:

```
sudo virsh pool-define-as --name default --type dir --target /var/lib/libvirt/images
sudo virsh pool-start default
sudo virsh pool-autostart default
```

7. Copy the Pull Secret file to the OpenShift install machine.

8. Set the environmental variables that OpenShift will use to grab the OpenShift install files:

```
export VERSION=stable-4.9
export RELEASE_IMAGE=$(curl -s https://mirror.openshift.com/pub/openshift-v4/clients/
ocp/$VERSION/release.txt | grep 'Pull From: quay.io' | awk -F ' ' '{print $3}')
export cmd=openshift-baremetal-install
export pullsecret_file=~/pull-secret.txt
export pullsecret_file=~/pull-secret
export extract_dir=$(pwd)
```

9. Download and unpack the OpenShift file:

```
curl -s https://mirror.openshift.com/pub/openshift-v4/clients/ocp/$VERSION/openshift-client-
linux.tar.gz | tar zxvf - oc
sudo cp oc /usr/local/bin
```

10. Using the OpenShift executable, download and extract the OpenShift installer, and move it to the bin directory:

```
oc adm release extract --registry-config "${pullsecret_file}" --command=$cmd --to "${extract_dir}"
${RELEASE_IMAGE}
sudo cp openshift-baremetal-install /usr/local/bin
```

Running an OpenShift install using installer-provided infrastructure

1. Create an install-config.yaml for your testbed (like the example below):

```
apiVersion: v1
baseDomain: [the domain you're using]
metadata:
  name: openshift
networking:
  machineCIDR: [a routable CIDR pool that can reach the Internet]
  networkType: OVNKubernetes
compute:
- name: worker
  replicas: 5
controlPlane:
  name: master
  replicas: 3
  platform:
    baremetal: {}
platform:
  baremetal:
    apiVIP: 10.209.60.201
    ingressVIP: 10.209.60.202
    provisioningNetwork: "Disabled"
    bootstrapProvisioningIP: 10.209.60.10
    hosts:
      - name: master-0
        role: master
        bmc:
          address: redfish-virtualmedia://10.209.101.11/redfish/v1/Systems/1
```

```
      username: Administrator
      password: Password1
      disableCertificateVerification: True
    bootMACAddress: [the MAC address of the NIC you're planning to use]
    hardwareProfile: default
    rootDeviceHints:
      model: "LOGICAL"
      minSizeGigabytes: 300
  - name: master-1
    role: master
    bmc:
      address: redfish-virtualmedia://10.209.101.12/redfish/v1/Systems/1
      username: Administrator
      password: Password1
      disableCertificateVerification: True
    bootMACAddress: [the MAC address of the NIC you're planning to use]
    hardwareProfile: default
    rootDeviceHints:
      model: "LOGICAL"
      minSizeGigabytes: 300
  - name: master-2
    role: master
    bmc:
      address: redfish-virtualmedia://10.209.101.13/redfish/v1/Systems/1
      username: Administrator
      password: Password1
      disableCertificateVerification: True
    bootMACAddress: [the MAC address of the NIC you're planning to use]
    hardwareProfile: default
    rootDeviceHints:
      model: "LOGICAL"
      minSizeGigabytes: 300
  - name: worker-0
    role: worker
    bmc:
      address: redfish-virtualmedia://10.209.101.21/redfish/v1/Systems/1
      username: Administrator
      password: Password1
      disableCertificateVerification: True
    bootMACAddress: [the MAC address of the NIC you're planning to use]
    rootDeviceHints:
      model: "LOGICAL"
      minSizeGigabytes: 300
  - name: worker-1
    role: worker
    bmc:
      address: redfish-virtualmedia://10.209.101.22/redfish/v1/Systems/1
      username: Administrator
      password: Password1
      disableCertificateVerification: True
    bootMACAddress: [the MAC address of the NIC you're planning to use]
```

```
        rootDeviceHints:
          model: "LOGICAL"
          minSizeGigabytes: 300
      - name: worker-2
        role: worker
        bmc:
          address: redfish-virtualmedia://10.209.101.23/redfish/v1/Systems/1
          username: Administrator
          password: Password1
          disableCertificateVerification: True
        bootMACAddress: [the MAC address of the NIC you're planning to use]
        rootDeviceHints:
          model: "LOGICAL"
          minSizeGigabytes: 300
      - name: worker-3
        role: worker
        bmc:
          address: redfish-virtualmedia://10.209.101.24/redfish/v1/Systems/1
          username: Administrator
          password: Password1
          disableCertificateVerification: True
        bootMACAddress: [the MAC address of the NIC you're planning to use]
        rootDeviceHints:
          model: "LOGICAL"
          minSizeGigabytes: 300
      - name: worker-4
        role: worker
        bmc:
          address: redfish-virtualmedia://10.209.101.25/redfish/v1/Systems/1
          username: Administrator
          password: Password1
          disableCertificateVerification: True
        bootMACAddress: [the MAC address of the NIC you're planning to use]
        rootDeviceHints:
          model: "LOGICAL"
          minSizeGigabytes: 300
pullSecret: '[your pull secret text]'
sshKey: '[your public SSH key]'
```

2. Create an install directory:

```
mkdir clusterconfig
```

3. Copy install-config.yaml into your install directory:

```
cp install-config.yaml clusterconfig/
```

4. To generate the Kubernetes manifests for your OpenShift cluster, run the following command:

```
openshift-baremetal-install --dir=clusterconfig create manifests
```

5. Create the ignition files:

```
openshift-baremetal-install --dir=clusterconfig create cluster
```

Creating the Local Storage operator

1. Log into your OpenShift Container Platform cluster.
2. Click Operators→OperatorHub.
3. In the search box, type `Local Storage`.
4. When the Local Storage operator appears, click it.
5. In the right side of the screen, click Install.
6. In the Create Operator Subscription screen, accept all defaults, and click Subscribe.
7. When the Installed Operators status page lists Local Storage as Succeeded, the installation of the operator is complete.

Creating the OpenShift Container Storage operator

1. Click Operators→OperatorHub.
2. In the search box, type `OpenShift Container Storage`.
3. When the OpenShift Container Storage operator appears, click it.
4. In the right side of the screen, click Install.
5. In the Create Operator Subscription screen, accept all defaults, and click Subscribe.

Creating an OpenShift Container Storage cluster

1. Click Operators→Installed Operators.
2. Click OpenShift Container Storage.
3. In OpenShift Container Storage, under the Storage Cluster option, click Create Instance.
4. In the Create Instance screen, make sure Internal-Attached devices is selected, select All Nodes, and click Next.
5. In the Create Storage Class screen, enter the Volume Set Name, and click Next.
6. When a pop-up warns you that this cannot be undone, click Yes to continue.
7. In the Set Storage and nodes screen, wait for the newly created Storage Class to finish creating, and click Next.
8. In the Security configuration screen, leave Enable encryption unchecked, and click Next.
9. To begin the OCS cluster creation in the Review screen, click Create.

Installing the OpenShift Virtualization operator

1. Click Operators→OperatorHub.
2. In the search box, type `OpenShift Virtualization`.
3. When the OpenShift Virtualization operator appears, click it.
4. In the right side of the screen, click Install.
5. In the Install Operator screen, accept all defaults, and click Subscribe.
6. After the Operator installs, the OpenShift Virtualization page will appear.
7. Click Create HyperConverged.
8. To add virtualization functionality to your cluster, in the Create HyperConverged screen, click Create.

Creating and configuring VMs for the two density tests

Creating a client VM

1. Click Workloads→Virtualization.
2. In Virtualization, click Create Virtual Machine.
3. In Select template, choose Microsoft Windows 10 VM, and click Next.
4. In boot source, choose a Windows 10 ISO, select This is a CD-ROM boot source, and click Customize virtual machine.
5. Name your VM.
6. Under Flavor, choose Large: 2 CPU| 8 GiB Memory, and click Next.
7. Leave networking settings at defaults, and click Next.
8. In Storage, create the OS disk (modified from `rootdisk`) with 30 GiB. Note: Be sure the disk is attaching to your OpenShift Container Storage classes.
9. Click Next.
10. In Advanced, click Next.
11. In Review, verify your configuration, and click Create Virtual Machine.
12. Click into the VM.
13. On the top right of the VM page, click the dropdown menu, and select Power On.
14. Open the console to the VM.

15. Mount the installation ISO to the VM, and restart it.
16. When prompted to boot from DVD, press any key.
17. When the installation screen appears, leave language, time/currency format, and input method as default, and click Next.
18. Click Install now.
19. When the installation prompts you, enter the product key.
20. Select Windows 10 Professional, and click Next.
21. Check I accept the license terms, and click Next.
22. Click Custom: Install Windows only (advanced).
23. Select Load Driver.
24. Navigate to the mounted OpenShift driver directory, and choose the appropriate Windows driver.
25. Select Drive 0 Unallocated Space, and click Next. Windows will begin and restart automatically.
26. When the Settings page appears, enter the same password in the Password and Reenter Password fields.
27. Log in with the password you set up in the previous step.
28. Using Windows update, download and install all updates.
29. To ensure the system does not apply new updates during testing, disable the Windows Update service.

Installing HammerDB 4.2

1. Download HammerDB from https://hammerdb.com/download.html.
2. Double-click the .exe file, choose English, and click OK.
3. Click Yes.
4. Click Next.
5. Chose the default destination location, and click Next.
6. Click Next.
7. Click Finish.

Creating a database VM

1. Click Workloads→Virtualization.
2. In Virtualization, click Create Virtual Machine.
3. In Select template, choose Microsoft Windows 10 VM, and click Next.
4. In boot source, choose a Windows 10 ISO, and select This is a CD-ROM boot source, and click Customize virtual machine.
5. Name your VM.
6. Under Flavor, choose Custom, 64 GiB Memory, 8 CPU, Workload Type as High-performance, and click Next.
7. Leave networking settings at defaults, and click Next.
8. In Storage, create the following disks (note: Be sure that the disks are attaching to your OpenShift Container Storage classes):
   - OS disk (modified from `rootdisk`): 50 GiB
   - Database disk: 100 GiB
   - Log disk: 50 GiB
9. Click Next.
10. In Advanced, click Next.
11. In Review, verify your configuration, and click Create Virtual Machine.
12. Click into the VM.
13. On the top right of the VM page, click the dropdown menu, and select Power On.
14. Open the console to the VM.
15. Mount the installation ISO to the VM, and restart it.
16. When prompted to boot from DVD, press any key.
17. When the installation screen appears, leave language, time/currency format, and input method as default, and click Next.
18. Click Install now.
19. When the installation prompts you, enter the product key.
20. Select Windows Server 2019 Standard Edition (Desktop Experience), and click Next.
21. Check I accept the license terms, and click Next.
22. Click Custom: Install Windows only (advanced).
23. Select Load Driver.
24. Navigate to the mounted OpenShift driver directory, and choose the appropriate Windows driver.
25. Select Drive 0 Unallocated Space, and click Next. Windows will begin and restart automatically.
26. When the Settings page appears, enter the same password in the Password and Reenter Password fields.

27. Log in with the password you set up in the previous step.
28. Right-click the Start Menu, and select Disk Management.
29. In Disk Management, right click the database drive, and select Online.
30. Right-click the database drive, and choose New Simple Volume.
31. In the New Volume screen, accept defaults, and click OK.
32. Complete steps 28 through 31 for the log volume.

Configuring Windows Server 2019

1. Open Server Manager, and click Local Server.
2. Disable IE Enhanced Security Configuration.
3. Change the time zone to your local time zone.
4. Change the name of your server. When prompted, reboot.
5. Open Server Manager again, and click Local Server.
6. Click the link to run updates.
7. Run updates, rebooting when prompted, until the server shows no new updates to install.

Installing SQL Server 2019 Enterprise

1. Download or copy the ISO to the server, and unzip it.
2. Double-click the Setup application.
3. Click Installation→New SQL Server Standalone installation or add features to an existing installation.
4. Choose the trial version, and click Next.
5. Select I accept the license terms and Privacy Statement, and click Next.
6. Select Use Microsoft Update to check for updates (recommended), and click Next.
7. On the Install Rules page, click Next.
8. Select the following features, and click Next:
   - Database Engine Services
   - Full-Test and Semantic Extractions for Search
   - Client Tools Connectivity
   - Client Tools Backwards Compatibility
9. Leave the default instance, and click Next.
10. Leave the default Service Accounts, and click Next.
11. On the Server Configuration tab, choose Mixed Mode, and enter and confirm a Password for the SQL Server system administrator (sa) account.
12. To specify the SQL Server administrators, click Add Current User.
13. Click Next.
14. Once you've passed the rule check, click Next.
15. Click Install.
16. When the install finishes, go back to the SQL Server Installation Center, and click Install SQL Server Management Tools.
17. Download the SSMS file, and install with defaults.
18. When prompted, reboot the server.
19. To ensure there aren't any new updates for SQL Server, run Windows Update one more time. Note: Be sure you have set Windows Updates to get updates for other Microsoft products.
20. Once you've installed all available updates, disable Windows Update service by performing the following actions:
    - Click Start.
    - To open the Services list, type `services`.
    - Disable the Windows Update service.

Locking pages in memory

1. Click Start, and type `Local Security Policy`.
2. When the program pops up in the search, open it.
3. Expand Local Policies, and click User Rights Assignment.
4. In the right pane, scroll down, and double-click Lock pages in memory.
5. Click Add User or Group, type `NT Service\MSSQLSERVER`, and click OK.
6. To close the Properties window, click OK.
7. Close the Local Security Policy window.

Configuring Microsoft SQL Server on the VMs under test

1. Open the SQL Server Management Studio.
2. Right-click Databases, and select New Database.
3. In the New Database screen, name your database, and click Filegroups.
4. In Filegroups, create eight database files on the database drive that are eight GB each, and one log file on the log drive that is 15 GB.
5. Click OK.

Populating the database

1. Open HammerDB on the client system, and click Options→Benchmark.
2. Choose Microsoft SQL and TPC-C.
3. Expand Schema Build, and double-click Options.
4. Set the IP of your SQL Server.
5. Choose SQL User (SA) connection, and enter the password.
6. Set the number of warehouses to 500.
7. Set the number of virtual users to 16.
8. Click OK.
9. Expand Virtual Users, and double-click Create.
10. To start the build, click Run Virtual Users.

Backing up the database

1. On the SQL server, open SQL Server Management Studio.
2. Right-click the TPC-C database, and click Tasks→Back up….
3. Select the database volume to store the backup file, enable compression, and click OK.

Performing the test

1. On the client system, start HammerDB.
2. Set the database server to SQL Server, and set the workload to TPC-C.
3. Open the Options panel for the Virtual Users: SQL Server→TPC-C→Virtual User→Options.
4. Type `16` for virtual users.
5. Select the following:
   - Show Output
   - Log Output to Temp
   - Use Unique Log Name
   - Log Timestamps
6. Click OK.
7. Open the Options panel for the Driver: SQL Server→TPC-C→Driver Script→Options.
8. Enter the IP of the target SQL Server.
9. Choose SQL Server Authentication connection, and enter the password.
10. Select Timed Driver Script, and type `15` for Rampup and `30` for the Test Duration.
11. Click Create Virtual Users.
12. To start the HammerDB workload, back on the client, click the green arrow.
13. When the run finishes, stop the data collector set on the target SQL Server.
14. Save the HammerDB results text file and data collector output for review.
15. Open SQL Manager, delete the TPCC database, and restore the database from the SQL backup file.
16. Reboot the SQL Server VM.

Resource management tasks

Performing the first VM density test (recording when performance drops below 50 percent)

This test assumes you have performed the VM installation steps for both the client and server VMs.

1. Perform a HammerDB test run with the following settings:
   - Warmup: 15 minutes
   - Test time: 30 minutes
   - Number of threads: 16
   - Warehouses: 500
2. Record the result.
3. Add pairs of SQL Server and client VMs, and perform the same HammerDB run on all pairs.
4. Record the result.
5. When the results are less than 50% of the performance of a single pair, record the number of pairs. Subtract one from this number to determine the number of pairs of VMs the solution supports.

Performing the second VM density test (recording how many idle database VMs a cluster can handle)

This test assumes you have performed the VM installation tests for the server VM. However, you must modify the server VM settings as follows:

- vCPUs: 2
- RAM: 16 GB
- Delete all disks besides the OS disk.

Clone out and power on VMs until the cluster cannot power on any more VMs, or it experiences instability upon powering on a new VM.

Load balancing VMs

This test assumes you have performed the VM installation tests for the server VM. However, you must modify the server VM settings as follows:

- RAM: 32 GB

Load balancing VMs on vSphere

1. Clone out VMs until there are five SQL Server VMs per VM host.
2. Record the names of the VMs that are running on Server 1.
3. On the VMs on Server 1, start a load test.

Load balancing VMs on OpenShift

1. Clone out VMs until there are five SQL Server VMs per VM host.
2. Record the names of the VMs that are running on Server 1.
3. On the VMs on Server 1, start a load test.
4. Once server 1 shows saturation, start manually migrating VMs from Server 1 to Server 2.

Management tasks

High availability

Enabling HA in a vSphere cluster

1. Click Menu→Hosts and Clusters.
2. Click the cluster you wish to use for HA.
3. Click the Configure tab.
4. Click vSphere Availability.
5. Beside vSphere HA is Turned OFF, click Edit.
6. Select vSphere HA, ensure that Enable Host Monitoring is also enabled, and click OK.

Enabling HA for an OpenShift cluster

1. Create a MachineHealthCheck yaml file. For reference, refer to the machinehealthcheck-bmh.yaml file in the Scripts we used in testing.
2. Apply the MachineHealthCheck:

```
oc create -f machinehealthcheck-bmh.yaml
```

Testing HA failover in a configured vSphere cluster

1. Write down the IP addresses of a VM running on Server 1.
2. Start the timer.
3. Unplug Server 1.
4. Start pinging the VM's IP address.
5. When the VM starts responding to pings, stop the timer.

Testing HA failover in a configured OpenShift cluster

1. Write down the IP addresses of a VM running on Server 1.
2. Start the timer.
3. Unplug Server 1.
4. Start pinging the VM's IP address.
5. When the VM starts responding to pings, stop the timer.

Live migration functionality

Live-migrating a VM to another host in vSphere

1. Click Menu→VMs and Templates.
2. Right-click the VM you wish to migrate, and select Migrate.
3. In the Migration menu, select Host migration only.
4. Choose the target host, and click Okay.

Live-migrating a VM to another host in OpenShift

1. In the Virtualization screen, select the VM you wish to migrate.
2. Select the menu on the right, and select Migrate Virtual Machine.

Migrating VM storage in vSphere

1. Right-click the VM, and select Migrate.
2. In the migration menu, select Storage migration only.
3. Select the target storage, and click Okay.

Migrating VM storage in OpenShift

OpenShift does not support live migrations of VM storage, but you can change the VM storage with the following steps:

1. In the OpenShift Container Platform console, navigate to Workloads→Virtualization.
2. To shut down the VM you wish you migrate, select the menu, and choose Stop Virtual Machine.
3. Click the name of the VM you wish to migrate.
4. In the VM menu, select Disks.
5. Record the name of the disk you wish to migrate.
6. Create a file to clone the disk you wish to migrate. To see the file we used to clone this disk, refer to clone-volume.yaml section Scripts we used in testing.
7. Run the disk cloning operation:

```
oc create -f clone-volume.yaml
```

8. Once the disk finishes cloning, navigate back to the VM menu, and select Add Disk.
9. In the Add Disk menu, choose the following options:
   - Source: [Use an existing PVC]
   - PersistentVolumeClaim: [The PVC on the new storage class you chose]
   - Name: [The name you want]
   - Type: Disk
   - Interface: virtio
10. Click Add.
11. To delete the original disk, choose the menu for the disk, and select Delete.
12. In the confirmation window, make sure Delete [your disk name] Data Volume and PVC is checked, and click Detach.
13. Click the Details tab.
14. Beside Boot Order in Details, click Edit.
15. Click Add Source.
16. Choose your new disk as the boot disk, and click Save.
17. Select Actions→Start Virtual Machine.

Enabling VMware EVC for live migration across heterogeneous hosts

1. Select Menu→Hosts and Clusters.
2. Right-click the cluster, and select Settings.
3. In the cluster settings screen, select VMware EVC.
4. In VMware EVC, click Edit.
5. In Change EVC Mode, select Enable EVC for Intel Hosts, choose the CPU generation you wish to enable, and click OK.

Testing VM functionality

Expanding a VM's virtual hardware in vSphere

Enabling VM virtual hardware expansion

1. Select Menu→VMs and Templates.
2. Right-click a powered-off VM to which you wish to enable hot-adding CPUs and memory, and select Edit Settings.
3. In Edit Settings, select Enable CPU Hot Add and Memory Hot Plug Enable, and click OK.
4. Power on the VM.

Expanding the hardware of a hot-add-enabled VM

1. Start the timer. (Because the VM technically never goes down, this is purely for bookkeeping purposes.)
2. Right-click the VM, and select Edit Settings.
3. In the VM settings, choose the new CPU and memory settings for your VM, and click Apply.
4. Stop the timer.

Expanding a VM's virtual hardware in OpenShift

1. Start the timer.
2. In the Virtualization screen, select the VM you wish to modify.
3. Select the menu on the right, then select Stop Virtual Machine.
4. To get to the overview screen once the VM stops, select the hyperlink to the VM.
5. In the VM overview screen, select the Details tab.
6. In the Details tab, click the pencil icon beside Flavor.
7. Choose your CPU and memory settings for your VM, and click Save.
8. Wait for the changes to propagate.
9. Select Actions→Start Virtual Machine.
10. When the VM is available again, stop the timer.

Taking a snapshot of a VM in vSphere

1. Select Menu→VMs and Templates.
2. Right-click the VM of which you wish to take a snapshot, and select Snapshots→Take Snapshot.
3. In the Take snapshot menu, select Create.

Taking a snapshot of a VM in OpenShift

1. In the Virtualization screen, select the VM you wish to snapshot.
2. Click the Snapshots tab.
3. Click Take Snapshot.
4. In the snapshots option screen, click Save.

Enabling VM encryption in vCenter

Creating a VM Encryption policy in vCenter

1. Click Menu→Policies and Profiles.
2. In VM Storage Policies, click Create.
3. In Create VM Storage Policy, name your new policy, and click Next.
4. In Policy structure, select Enable host based rules and Enable rules for vSAN storage, and click Next.
5. In Host-based services, choose Custom, choose VMware VM Encryption, and click Next.
6. In vSAN, leave the defaults (None - standard cluster and 1 failure - RAID-1 (Mirroring)), and click Next.
7. In Storage compatibility, verify that your datastore is compatible, and click Next.
8. In Review and finish, verify your settings, and click Finish.

Creating a Native Key Provider in vCenter

1. Click Menu→Hosts and Clusters.
2. Click the vCenter.
3. On the right side, click Configure.
4. In the Configure window, click Key Providers.
5. In Key Providers, click Add→Native Key Provider.
6. Name your Native Key Provider, and click Add Key Provider.
7. Click your newly created Native Key Provider, and click Back Up.
8. Select Protect Native Key Provider data with password (recommended).
9. Enter your password, select I have saved the password in a secure place, and click Back Up Key Provider.
10. When prompted, save the backup to your local machine.

Adding at-rest encryption to OpenShift Data Foundation

Installing and configuring a Key Management Service (KMS) for OpenShift

1. Install HashiCorp Vault to the system you will use as your KMS:

```
sudo yum install -y yum-utils
sudo yum-config-manager --add-repo https://rpm.releases.hashicorp.com/RHEL/hashicorp.repo
sudo yum -y install vault
```

2. Configure Vault:

```
vi vault-server-hcl [paste the contents of the vault_server_hcl file into this file, making sure to change IPs and hostnames as appropriate]
mkdir -p ./vault/data
mkdir -p ./vault/config
```

3. For more information on the vault-server-hcl file, see the section Scripts we used in testing.
4. Start HashiCorp Vault:

```
vault server -config ./vault/config/vault-server-hcl
export VAULT_ADDR="http://$(hostname):8200"
```

5. Initialize HashiCorp Vault:

```
vault operator init
```

6. Record the five unseal keys and the initial root token while completing steps 7 through 9.
7. Unseal HashiCorp Vault by typing the following command three times (note: When prompted, add a different unseal key each time):

```
vault operator unseal
```

8. For future configuration, enable username and password logins:

```
vault login [the root token you recorded earlier]
vault auth enable userpass
vault write auth/userpass/users/vaultuser password='Password1!' policies=admins
```

9. For the KMS, create a dedicated KV store:

```
vault secrets enable -path=ocs kv
echo 'path "ocs/*" {
  capabilities = ["create", "read", "update", "delete", "list"]
}
  path "sys/mounts" {
  capabilities = ["read"]
 }'| vault policy write ocs -
vault token create -policy=ocs -format json
```

10. Record the client_token from the Vault output. OpenShift Data Foundation will use this in later steps.

Adding encrypted StorageClasses to OpenShift Data Foundation

1. In the OpenShift console, select Storage→StorageClasses.
2. In StorageClasses, select Create StorageClass.
3. Select the following options for your StorageClass:
   - Name: `encrypted-sc`
   - Description: [leave blank]
   - Reclaim policy: Delete
   - Volume binding mode: WaitForFirstConsumer
   - Provisioner: openshift-storage.rbd.csi.ceph.com
   - Storage Pool: ocs-storagecluster-cephblockpool
   - Check Enable Encryption
   - Create New KMS Connection
   - Service name: `vault-kms`
   - Address: `http://provisioner.openshift.testbed.local`
   - Port: `8200`
   - Click Advanced Settings.
     - Backend Path: ocs
     - TLS Server Name: provisioner.openshift.testbed.local
     - Click Save.
   - In Connect to a Key Management Service, click Save.
   - Uncheck Allow PersistentVolumeClaims to be expanded
4. Click Create.

Configuring namespaces using the encrypted StorageClass

1. For any namespace you want to enable to use the encrypted StorageClass, you must add the following secret to the namespace:

```
---
apiVersion: v1
kind: Secret
metadata:
  name: ceph-csi-kms-token
  namespace: my-rbd-storage
stringData:
  token: "s.v8r8z4HfnboRsArn0LnjYdUs"
```

2. Apply the secret to the namespace:

```
oc create -f kms-secret.yaml
```

Scripts we used in testing

machinehealthcheck-bmh.yaml

```
apiVersion: machine.openshift.io/v1beta1
kind: MachineHealthCheck
metadata:
  name: methodology-mhc
  namespace: openshift-machine-api
  annotations:
    machine.openshift.io/remediation-strategy: external-baremetal
spec:
  selector:
    matchLabels:
      machine.openshift.io/cluster-api-machine-role: worker
      machine.openshift.io/cluster-api-machine-type: worker
      machine.openshift.io/cluster-api-machineset: openshift-t6zc9-worker-0
  unhealthyConditions:
    - type: Ready
      status: Unknown
      timeout: 300s
    - type: Ready
      status: 'False'
      timeout: 300s
  nodeStartupTimeout: "60m"
```

vault-server-hcl

```
disable_mlock = true
ui = true
listener "tcp" {
  address = "{ip_to_bind_to}:8200"
  tls_disable = "true"
}

cluster_name = "localvault"
api_addr = "http://provisioner.openshift.testbed.local:8200"
cluster_addr = "http://provisioner.openshift.testbed.local:8201"
storage "file" {
  path = "./vault/data"
}
```

clonevolume.yaml

```
apiVersion: cdi.kubevirt.io/v1beta1
kind: DataVolume
metadata:
  namespace: "openshift-cnv"
  name: ws-2019-os-gold
spec:
  source:
    pvc:
      namespace: "openshift-cnv"
      name: "win2k19-official-snake-rootdisk-fzto1"
  pvc:
    accessModes:
      - ReadWriteMany
    storageClassName: virtual-machine-2-replicas
    volumeMode: Block
    resources:
      requests:
        storage: 40Gi
```

Read the report at https://facts.pt/Nc0jS6Q ▶

This project was commissioned by VMware.

Principled Technologies®

Facts matter.®