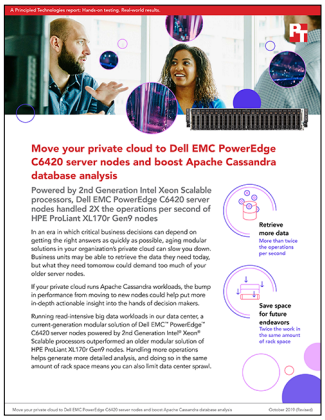

The science behind the report:

Move your private cloud to Dell EMC PowerEdge C6420 server nodes and boost Apache Cassandra database analysis

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 Move your private cloud to Dell EMC PowerEdge C6420 server nodes and boost Apache Cassandra database analysis.

We concluded our hands-on testing on September 4, 2019. 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 August 28, 2019 or earlier. Unavoidably, these configurations may not represent the latest versions available when this report appears.

Our results

	Dell EMC™ PowerEdge™ C6420 solution	HPE ProLiant XL170r Gen9 solution	Percentage win
Operations per second	353,063	161,922	118%

System configuration information

The table below presents detailed information on the systems we tested.

Server configuration information

Server configuration information	Dell EMC PowerEdge C6420	HPE ProLiant XL170r Gen9
BIOS name and version	Dell EMC 2.2.1	HPE U14 v2.72
Operating system name and version/build number	VMware ESXi™ 6.7.0 Update 3 build-14320388	VMware ESXi 6.7.0 Update 3 build-14320388
Date of last OS updates/patches applied	08/28/19	08/28/19
Power management policy	Performance	Static High Performance Mode
Processor		
Number of processors	2	2
Vendor and model	Intel® Xeon® Gold 6230	Intel Xeon E5-2698 v4
Core count (per processor)	20	20
Core frequency (GHz)	2.10	2.20
Stepping	6	B0
Memory module(s)		
Total memory in system (GB)	192	192
Number of memory modules	12	12
Vendor and model	Micron® MTA18ASF2G72PDZ-2G9E1TK	Hynix® HMA81GR7AFR8N
Size (GB)	16	16
Type	PC4-23466	PC4-21333
Speed (MHz)	2,933	2,600
Speed running in the server (MHz)	2,933	2,133
Storage controller		
Vendor and model	Dell PERC H330 Mini	HPE Smart Array P440 Controller
Cache size (GB)	0	4
Firmware version	25.5.5.0005	6.88
Local storage		
Number of drives	6	6
Drive vendor and model	Intel SSDSC2KB019T8R	HPE EH0600JEDHE
Drive size (GB)	1,920	600
Drive information (speed, interface, type)	6Gb, SAS, SSD	15K, 12Gb SAS, HDD
Network adapter		
Vendor and model	Intel XXV710	HPE 560FLR-SFP+
Number and type of ports	2 x 25GbE	2 x 10GbE
Driver version	I40en 1.8.1.9-2vmw.670.3.73.14320388	Ixgben 1.7.10-1OEM.670.0.0.8169922

Chassis configuration information

Chassis configuration information	Dell EMC PowerEdge C6400	HPE Apollo r2600 Gen9
Cooling fans		
Vendor and model number	Delta GFC0612DS	HPE 800059-B21
Number of cooling fans	4	4
Power supplies		
Vendor and model	Dell EMC D2000E-S0	HPE 720479-B21
Number of power supplies	2	2
Wattage of each (W)	2,000	800

How we tested

Configuring storage

We configured the HPE ProLiant XL170r Gen9 server nodes with six hard disk drives (HDDs) and the Dell EMC PowerEdge C6420 server nodes with six solid-state drives (SSDs). We then created a RAID5 array in the BIOS of each server using the six drives in each and default RAID controller settings.

Installing VMware ESXi 6.7

1. Attach the installation media.
2. Boot the server.
3. At the VMware Installer screen, press Enter.
4. At the EULA screen, to Accept and Continue, press F11.
5. Under Storage Devices, select the appropriate virtual disk, and press Enter.
6. As the keyboard layout, select US, and press Enter.
7. Enter the root password twice, and press Enter.
8. To start installation, press F11.
9. After the server reboots, press F2, and enter root credentials
10. Select Configure Management Network, and press Enter.
11. Select the appropriate network adapter, and select OK.
12. Select IPv4 settings, and enter the desired IP address, subnet mask, and gateway for the server.
13. Select OK, and restart the management network.
14. For the rest of the servers, complete steps 1 through 13.

Deploying the VMware vCenter® Server 6.7

1. On a Microsoft Windows server or VM, locate the VMware-VCSA installer image.
2. Mount the image, navigate to the vcsa-ui-installer folder, and double-click win32.
3. Double-click installer.exe.
4. Click Install.
5. Click Next.
6. Accept the terms of the license agreement, and click Next.
7. Leave the default vCenter Server with an Embedded Platform Services Controller selected, and click Next.
8. Enter the FQDN or IP address of the host for the vCenter Server Appliance deployment.
9. Provide the server's username and password, and click Next.
10. Accept the certificate of the host you chose to connect to by clicking Yes.
11. Provide a name and password for the vCenter Appliance, and click Next.
12. Set an appropriate Appliance Size, and click Next.
13. Select the appropriate datastore, and click Next.
14. At the Configure Network Settings page, configure the network settings as appropriate for your environment, and click Next.
15. Review your settings, and click Finish.
16. When the deployment completes, click Next.
17. At the Introduction page, click Next.
18. At the Appliance configuration page, select the time synchronization mode and SSH access settings, and click Next.
19. Select Create a new SSO domain.
20. Provide a password, and confirm it.
21. Provide an SSO Domain name and SSO Site name, and click Next.
22. At the CEIP page, click Next.
23. At the Ready to complete page, click Finish.
24. When installation completes, click Close.
25. Using the VMware vSphere® web client, log into the vCenter server using the credentials previously provided.

Creating a cluster and adding the hosts to VMware vCenter

1. After logging into vCenter, navigate to Hosts and Clusters.
2. Select the primary site management vCenter.
3. Right-click the vCenter, and select New Datacenter...
4. Enter a name for the new data center, and click OK.
5. Right-click the new data center, and click New Cluster...
6. Type `Dell Cluster` and `HPE Cluster` as the names for the new clusters.
7. Click OK.

Configuring networking

On each host in the HPE ProLiant XL170r Gen9 or Dell EMC PowerEdge C6420 cluster, we configured one 1Gb network interface with a VMKernel NIC set up for management traffic. We then configured one 10Gb/25Gb network interface with a distributed vSwitch for VM network traffic. We configured the second 10Gb/25Gb network interface with another distributed vSwitch and a VMKernel NIC interface set up for vMotion/Provisioning traffic. We enabled jumbo frames on all 10Gb/25Gb network interfaces and switches. This allowed plenty of bandwidth for the Yahoo Cloud Serving Benchmark (YCSB) workload as well as high availability for the clusters created in vCenter. We also configured a single 10Gb network interface on the infrastructure Dell EMC PowerEdge R740 for a management traffic VMKernel NIC and a VM Network virtual port group.

Creating the workload virtual machines

We created varying numbers of workload VMs for our testing depending on the generation under test. We used 10 VMs per server. Each VM had eight vCPUs and 16 GB of memory, and a connection to our testing network.

Creating the base VM

1. In VMware vCenter, navigate to Virtual Machines.
2. Click Create a new VM.
3. Leave Create a new virtual machine selected, and click Next.
4. Enter a name for the virtual machine, and click Next.
5. Place the VM on a host with available CPUs, and click Next.
6. Select the first the 8GB OS VMDK, and click Next.
7. Click Next.
8. Select the guest OS as Red Hat® Enterprise Linux® 8, and click Next.
9. In the Customize Hardware section, make the following changes:
    a. Increase the vCPUs to eight.
    b. Increase the VM memory to 16 GB.
    c. Add a 32GB VMDK for Cassandra data, and select the VMware Paravirtual controller and Thick Provisioned Eager Zeroed.
    d. Add a 16GB VMDK for Cassandra commitlog, and select the VMware Paravirtual controller and Thick Provisioned Eager Zeroed.
10. Connect the VM to the test network.
11. Click Next.
12. Click Finish.

Installing Red Hat Enterprise Linux 8

1. Attach the Red Hat Enterprise Linux 8 installation media, and boot the VM.
2. Select Install Red Hat Enterprise Linux 8.0.
3. Choose a language, and click Continue.
4. Click Software Selection. For the Base Environment, select Minimal. Under Add-Ons, check Guest Agents. To exit Software Selection, click Done.
5. Click Network & Host Name. Set a Host Name, and click Apply. Click Configure, and assign a static IP address. To exit Network & Host Name screen, click Done.
6. Click Installation Destination.
   a. Select Custom under Storage Configuration. Click Done.
   b. From the drop-down menu, select Standard Partition.
   c. To create a new partition, click +. Select /boot/efi as the mount point, and set the capacity as 600M. Click Add mount point.
   d. To create a new partition, click +. Select / as the mount point, and leave the capacity field empty. Click Add mount point.
   e. Click Done. To ignore the warning about missing a swap partition, click Done. Click Accept Changes.
   f. At the installation welcome screen, press Next.
7. Click Begin Installation.
8. During the installation, click Root Password. Enter a root password twice, and click Done. If you get a warning, click Done.
9. To reboot the VM after the installation completes, click Done.

Preparing the Apache Cassandra™ VM

1. Boot the Red Hat Enterprise Linux 8 VM that will run Cassandra.
2. Log in as root.
3. Disable SELinux:

```
setenforce 0
sed -i 's/SELINUX=.*/SELINUX=disabled/' /etc/selinux/config
```

4. Disable the firewall:

```
systemctl stop firewalld
systemctl mask firewalld
```

5. Configure SSH to work without a password:
   a. Run the following commands:

```
rm -rf ~/.ssh
mkdir -p ~/.ssh
chmod 700 ~/.ssh
cd ~/.ssh
```

   b. For any prompts with the following command, press Enter:

```
ssh-keygen -t rsa -q
```

   c. Run these commands:

```
cp id_rsa.pub authorized_keys
echo "StrictHostKeyChecking=no" > config
echo > known_hosts
```

6. Install the latest updates, and reboot:

```
dnf update -y
reboot
```

7. Install basic tools:

```
dnf install -y tuned wget vim-common vim openssh-clients sysstat
```

8. To configure the /etc/sysctl.conf file, run the following command:

```
cat >> /etc/sysctl.conf <<EOF
net.ipv4.tcp_keepalive_time=60
net.ipv4.tcp_keepalive_probes=3
net.ipv4.tcp_keepalive_intvl=10
net.core.rmem_max=16777216
net.core.wmem_max=16777216
net.core.rmem_default=16777216
net.core.wmem_default=16777216
net.core.optmem_max=40960
net.ipv4.tcp_rmem=4096 87380 16777216
net.ipv4.tcp_wmem=4096 65536 16777216
vm.max_map_count = 1048575
EOF
```

9. From Scripts we used while preparing the Cassandra VM, add the script file `/etc/rc.d/rc.local`.
10. From Scripts we used while preparing the Cassandra VM, add the template file `/etc/hosts_template`.
11. To enable the template, run the following commands:

```
chmod +x /etc/rc.d/rc.local
systemctl enable rc-local
grub2-mkconfig -o /boot/efi/EFI/redhat/grub.cfg
```

Installing Cassandra

1. Install the prerequisite software (Java and Python):

```
dnf install -y java-1.8.0-openjdk python2
alternatives --set python /usr/bin/python2
```

2. Configure Cassandra repo:

```
cat > /etc/yum.repos.d/cassandra.repo <<EOF
[cassandra]
name=Apache Cassandra
baseurl=https://www.apache.org/dist/cassandra/redhat/311x/
gpgcheck=1
repo_gpgcheck=1
gpgkey=https://www.apache.org/dist/cassandra/KEYS
EOF
```

3. Install Cassandra:

```
dnf install -y cassandra
```

4. Configure Cassandra service, and disable autostart:

```
cat > /etc/systemd/system/cassandra.service <<EOF
[Unit]
Description=Apache Cassandra
After=network.target

[Service]
PIDFile=/var/run/cassandra/cassandra.pid
User=cassandra
Group=cassandra
ExecStart=/usr/sbin/cassandra -f -p /var/run/cassandra/cassandra.pid
Restart=always

[Install]
WantedBy=multi-user.target
EOF

systemctl daemon-reload
systemctl stop cassandra
systemctl disable cassandra
```

5. Configure storage for Cassandra:

```
systemctl stop cassandra
rm -rf /var/lib/cassandra/*

DATA_DEV=/dev/sdb
COMMIT_DEV=/dev/sdc
DATA_DIR=/var/lib/cassandra
COMMIT_DIR=/var/lib/cassandra/commitlog

mkfs.xfs -L cass_data ${DATA_DEV}
mkfs.xfs -L cass_commit ${COMMIT_DEV}

mkdir -p ${DATA_DIR}
echo -e "UUID=$(blkid -o value -s UUID ${DATA_DEV})  ${DATA_DIR}  xfs  noatime,nodiratime,x-systemd.
device-timeout=10,nofail 0 0" >> /etc/fstab
mount -v ${DATA_DIR}

mkdir -p ${COMMIT_DIR}
echo -e "UUID=$(blkid -o value -s UUID ${COMMIT_DEV})  ${COMMIT_DIR}  xfs  noatime,nodiratime,x-
systemd.device-timeout=10,nofail 0 0" >> /etc/fstab
mount -v ${COMMIT_DIR}

chown cassandra:cassandra ${DATA_DIR} ${COMMIT_DIR}
```

6. Configure Cassandra config file:

```
vi /etc/cassandra/conf/cassandra.yaml
```

   a. Set `seeds = HOST_NAME_OF_SEED_VMs`, using 1 seed VM from each server, separated by commas.
   b. Set `data_directory = /PATH/TO/DATAMOUNT`
   c. Set `commitlog_directory = /PATH/TO/COMMITLOGMOUNT`
   d. Set `listen_interface = VM_INTERFACE_NAME`
   e. Set `rpc_interface = VM_INTERFACE_NAME`
   f. Set `endpoint_snitch = GossipingPropertyFileSnitch`
   g. Set `compaction_throughput_mb_per_sec = 0`
   h. Set `trickle_fsync = true`
   i. Set `disk_optimization_strategy = spinning OR ssd`

7. Reboot the VM, and start Cassandra. Then verify Cassandra is working (Cassandra may take a minute to start):

```
reboot
systemctl start cassandra
nodetool status
```

8. Power off this VM, and clone it as many times as you need to so each cluster has 10 VMs per server (40 VMs for our solutions).
9. To add new VMs to each cluster, boot them individually. To see the progress of the new VMs joining each Cassandra cluster, run `nodetool status` from the first VM in each cluster.

Installing YCSB

Clone a Red Hat Enterprise Linux 8 VM onto the Dell EMC PowerEdge R740 infrastructure host. Each YCSB driver VM will reside there.

1. Boot a Red Hat Enterprise Linux 8 VM for YCSB.
2. Provide a static IP address, gateway, and DNS server, and restart the networking service.
3. Download YCSB 0.15.0.
4. Power off this VM, and clone it out for a total of four YCSB driver VMs.
5. Boot them, and provide each with a unique static IP address.

Creating a Cassandra table for use with YCSB

1. From one of the Cassandra nodes, log into Cassandra Command Line Shell (cqlsh):

   ```
   cqlsh $HOSTNAME
   ```

2. Create a YCSB keyspace with a network topology replication strategy and replication factor equal to three (replace `DC_NAME` with your cluster data center's name set in the cassandra.conf file):

   ```
   cqlsh> create keyspace ycsb WITH REPLICATION = {'class' : 'NetworkTopologyStrategy', 'DC_NAME': 3 };
   ```

3. In the YCSB keyspace, create a table named `usertable` with 10 columns:

   ```
   cqlsh> USE ycsb;
   cqlsh> create table usertable (
       y_id varchar primary key,
       field0 varchar,
       field1 varchar,
       field2 varchar,
       field3 varchar,
       field4 varchar,
       field5 varchar,
       field6 varchar,
       field7 varchar,
       field8 varchar,
       field9 varchar)
       WITH compaction = { 'class' :  'LeveledCompactionStrategy'  };
   ```

4. Populate `usertable` from a YCSB driver VM:

   ```
   /root/ycsb-0.15.0/bin/ycsb load cassandra-cql -p hosts=IP_ADDRESS_OF_CASSANDRA_NODES_TO_CONNECT_TO
   -threads NUMBER_OF_THREADS -p fieldcount=10 -p recordcount=RECORD_COUNT -p  -P /root/ycsb-0.15.0/
   workloads/workloadb
   ```

Performing a YCSB run on the Cassandra Database

1. From a YCSB driver VM, run the following command:

   ```
   ./bin/ycsb run cassandra-cql -p hosts=IP_ADDRESS_OF_CASSANDRA_NODES_TO_CONNECT_TO -threads NUMBER_
   OF_THREADS -p fieldcount=10 -p operationcount=OPERATION_COUNT -p recordcount=RECORD_COUNT -p
   requestdistribution=DISTRIBUTION_MODEL -P /root/ycsb-0.15.0/workloads/workloadb > YCSB_results_
   workloadb
   ```

Testing the process

1. Create a 60GB Cassandra database (recordcount=60000000):

   ```
   ./bin/ycsb load cassandra-cql -p hosts="vm-r1-n1 vm-r2-n1 vm-r3-n1 cm-r4-n1" -threads 60 -p
   fieldcount=10 -p recordcount=60000000 -p requestdistribution=zipfian -P workloads/workloadb
   ```

2. Run the YCSB benchmark against the Cassandra database with four driver VMs. In our testing, we started the measurement with 40 YCSB threads (10 on each YCSB driver VM) against the Cassandra database. The YCSB threads connect to the IP addresses of four Cassandra VMs (the first VM and seed node on each physical server). For reads and writes in our testing, we used a QUORUM consistency level. We set our max runtime to 20 minutes (1,200 seconds). For each test run, use YCSB/workload-b (95% reads and 5% updates) and the Zipfian distribution model:

   ```
   ./bin/ycsb run cassandra-cql -p hosts="vm-r1-n1 vm-r2-n1 vm-r3-n1 cm-r4-n1" -threads thread_count
   -p fieldcount=10 -p operationcount=600000000 -p recordcount=60000000 -p requestdistribution=zipfian
   -P workloads/workloadb -p maxexecutiontime=1200 -p readconsistencylevel=QUORUM -p
   writeconsistencylevel=QUORUM
   ```

3. While the test is in progress, monitor and collect ESXTOP statistics (CPU utilization, disk IO latency, network throughput, and memory usage) from the servers.
4. After the test completes, record the overall operations per second.
5. In the subsequent runs, increase the YCSB threads until performance no longer improves. In our tests, we performed the final test runs with 480 threads total (120 per driver).
6. From the backup, delete and restore the database files between each run.
7. Repeat the process to total three test runs, and record the median operations per second.

Scripts we used while preparing the Cassandra VM

To prepare the VM for Cassandra installation, use the following script in step 9 of the Preparing Apache Cassandra VM section in the methodology.

```
#!/bin/bash

touch /var/lock/subsys/local

CASS_CONFDIR=/etc/cassandra/conf

DOMAIN="test.local"
IFNAME="ens192"

export DOMAIN IFNAME

ETHER='ip addr show dev ${IFNAME} | awk '/ether/{print $2;exit}''
echo ${ETHER} > /run/ether
SUBNETVAR='awk -F':' '{printf("%d",strtonum("0x"$1))}' /run/ether'
RACK='awk -F':' '{printf("%d",strtonum("0x"$2))}' /run/ether'
NODE='awk -F':' '{printf("%d",strtonum("0x"$3))}' /run/ether'
HOSTPREFIX='awk -F':' '{print $4$5$6}' /run/ether | xxd -r -p'

export ETHER SUBNETVAR RACK NODE HOSTPREFIX

echo ${DOMAIN} > /run/domain
echo ${IFNAME} > /run/ifname
echo ${SUBNETVAR} > /run/subnetvar
echo ${RACK} > /run/rack
echo ${NODE} > /run/node
echo ${HOSTPREFIX} > /run/hostprefix

NEW_HOSTNAME="${HOSTPREFIX}-r${RACK}-n${NODE}"
export NEW_HOSTNAME
hostnamectl set-hostname ${NEW_HOSTNAME}.${DOMAIN}

REBOOT=0
# Hosts config
envsubst < /etc/hosts_template > /etc/hosts_new
if [ "$(diff -q /etc/hosts_new /etc/hosts)" != "" ]; then
cp -pf /etc/hosts_new /etc/hosts
REBOOT=1
fi

# NIC config
cat > /etc/ifcfg-${IFNAME}_new <<EOF
TYPE=Ethernet
BOOTPROTO=none
IPV6INIT=no
ONBOOT=yes
PREFIX=16 MTU=9000
NAME=${IFNAME}
DEVICE=${IFNAME}
IPADDR=10.${SUBNETVAR}.${RACK}.${NODE}
GATEWAY=10.${SUBNETVAR}.0.1
DNS1=10.${SUBNETVAR}.0.1
DOMAIN=${DOMAIN}
EOF
if [ "$(diff -q /etc/ifcfg-${IFNAME}_new /etc/sysconfig/network-scripts/ifcfg-${IFNAME})" != "" ]; then
cp -pf /etc/ifcfg-${IFNAME}_new /etc/sysconfig/network-scripts/ifcfg-${IFNAME}
REBOOT=1
fi

# Cassandra config
```

```
cp -pf ${CASS_CONFDIR}/cassandra-rackdc.properties{,_backup}
sed -i -e "s/rack=.*/rack=rack${RACK}/" -e "s/dc=.*/dc=dc${SUBNETVAR}/" ${CASS_CONFDIR}/cassandra-rackdc.
properties

cp -pf ${CASS_CONFDIR}/cassandra.yaml{,_backup}
sed -i "s/cluster_name:.*/cluster_name: '${HOSTPREFIX} cluster'/" ${CASS_CONFDIR}/cassandra.yaml
if [ "$(diff -q ${CASS_CONFDIR}-rackdc.properties{,_backup})" != "" ] ; then
REBOOT=1
fi


# Reboot if needed
if [ 0${REBOOT} -ne 0 ]; then
if [ ! -e /noreboot ]; then
echo REBOOT REQUIRED!
systemctl stop cassandra
sleep 1
rm -rf /var/lib/cassandra/{commitlog,data,hints,saved_caches}
rm -rf /var/log/cassandra/*
sync
fstrim -av
echo > /root/.ssh/known_hosts
sleep 3
reboot
fi
fi

# Try to extend filesystem
sleep 1
xfs_growfs /var/lib/cassandra
xfs_growfs /var/lib/cassandra/commitlog

# Set readahead low
blockdev --setra 128 /dev/sd[a-z]

exit 0
```

To prepare the VM for Cassandra installation, use the following script in step 10 of the "Preparing Cassandra VM" section in the methodology.

```
127.0.0.1      localhost localhost.localdomain localhost4 localhost4.localdomain4
::1    localhost localhost.localdomain localhost6 localhost6.localdomain6

10.${SUBNETVAR}.1.1    ${HOSTPREFIX}-r1-n1.${DOMAIN} ${HOSTPREFIX}-r1-n1 cass-r1-n1
10.${SUBNETVAR}.1.2    ${HOSTPREFIX}-r1-n2.${DOMAIN} ${HOSTPREFIX}-r1-n2 cass-r1-n2
10.${SUBNETVAR}.1.3    ${HOSTPREFIX}-r1-n3.${DOMAIN} ${HOSTPREFIX}-r1-n3 cass-r1-n3
10.${SUBNETVAR}.1.4    ${HOSTPREFIX}-r1-n4.${DOMAIN} ${HOSTPREFIX}-r1-n4 cass-r1-n4
10.${SUBNETVAR}.1.5    ${HOSTPREFIX}-r1-n5.${DOMAIN} ${HOSTPREFIX}-r1-n5 cass-r1-n5
10.${SUBNETVAR}.1.6    ${HOSTPREFIX}-r1-n6.${DOMAIN} ${HOSTPREFIX}-r1-n6 cass-r1-n6
10.${SUBNETVAR}.1.7    ${HOSTPREFIX}-r1-n7.${DOMAIN} ${HOSTPREFIX}-r1-n7 cass-r1-n7
10.${SUBNETVAR}.1.8    ${HOSTPREFIX}-r1-n8.${DOMAIN} ${HOSTPREFIX}-r1-n8 cass-r1-n8
10.${SUBNETVAR}.1.9    ${HOSTPREFIX}-r1-n9.${DOMAIN} ${HOSTPREFIX}-r1-n9 cass-r1-n9
10.${SUBNETVAR}.1.10   ${HOSTPREFIX}-r1-n10.${DOMAIN} ${HOSTPREFIX}-r1-n10 cass-r1-n10

10.${SUBNETVAR}.2.1    ${HOSTPREFIX}-r2-n1.${DOMAIN} ${HOSTPREFIX}-r2-n1 cass-r2-n1
10.${SUBNETVAR}.2.2    ${HOSTPREFIX}-r2-n2.${DOMAIN} ${HOSTPREFIX}-r2-n2 cass-r2-n2
10.${SUBNETVAR}.2.3    ${HOSTPREFIX}-r2-n3.${DOMAIN} ${HOSTPREFIX}-r2-n3 cass-r2-n3
10.${SUBNETVAR}.2.4    ${HOSTPREFIX}-r2-n4.${DOMAIN} ${HOSTPREFIX}-r2-n4 cass-r2-n4
10.${SUBNETVAR}.2.5    ${HOSTPREFIX}-r2-n5.${DOMAIN} ${HOSTPREFIX}-r2-n5 cass-r2-n5
10.${SUBNETVAR}.2.6    ${HOSTPREFIX}-r2-n6.${DOMAIN} ${HOSTPREFIX}-r2-n6 cass-r2-n6
10.${SUBNETVAR}.2.7    ${HOSTPREFIX}-r2-n7.${DOMAIN} ${HOSTPREFIX}-r2-n7 cass-r2-n7
10.${SUBNETVAR}.2.8    ${HOSTPREFIX}-r2-n8.${DOMAIN} ${HOSTPREFIX}-r2-n8 cass-r2-n8
10.${SUBNETVAR}.2.9    ${HOSTPREFIX}-r2-n9.${DOMAIN} ${HOSTPREFIX}-r2-n9 cass-r2-n9
10.${SUBNETVAR}.2.10   ${HOSTPREFIX}-r2-n10.${DOMAIN} ${HOSTPREFIX}-r2-n10 cass-r2-n10

10.${SUBNETVAR}.3.1    ${HOSTPREFIX}-r3-n1.${DOMAIN} ${HOSTPREFIX}-r3-n1 cass-r3-n1
10.${SUBNETVAR}.3.2    ${HOSTPREFIX}-r3-n2.${DOMAIN} ${HOSTPREFIX}-r3-n2 cass-r3-n2
```

```
10.${SUBNETVAR}.3.3    ${HOSTPREFIX}-r3-n3.${DOMAIN} ${HOSTPREFIX}-r3-n3 cass-r3-n3
10.${SUBNETVAR}.3.4    ${HOSTPREFIX}-r3-n4.${DOMAIN} ${HOSTPREFIX}-r3-n4 cass-r3-n4
10.${SUBNETVAR}.3.5    ${HOSTPREFIX}-r3-n5.${DOMAIN} ${HOSTPREFIX}-r3-n5 cass-r3-n5
10.${SUBNETVAR}.3.6    ${HOSTPREFIX}-r3-n6.${DOMAIN} ${HOSTPREFIX}-r3-n6 cass-r3-n6
10.${SUBNETVAR}.3.7    ${HOSTPREFIX}-r3-n7.${DOMAIN} ${HOSTPREFIX}-r3-n7 cass-r3-n7
10.${SUBNETVAR}.3.8    ${HOSTPREFIX}-r3-n8.${DOMAIN} ${HOSTPREFIX}-r3-n8 cass-r3-n8
10.${SUBNETVAR}.3.9    ${HOSTPREFIX}-r3-n9.${DOMAIN} ${HOSTPREFIX}-r3-n9 cass-r3-n9
10.${SUBNETVAR}.3.10   ${HOSTPREFIX}-r3-n10.${DOMAIN} ${HOSTPREFIX}-r3-n10 cass-r3-n10

10.${SUBNETVAR}.4.1    ${HOSTPREFIX}-r4-n1.${DOMAIN} ${HOSTPREFIX}-r4-n1 cass-r4-n1
10.${SUBNETVAR}.4.2    ${HOSTPREFIX}-r4-n2.${DOMAIN} ${HOSTPREFIX}-r4-n2 cass-r4-n2
10.${SUBNETVAR}.4.3    ${HOSTPREFIX}-r4-n3.${DOMAIN} ${HOSTPREFIX}-r4-n3 cass-r4-n3
10.${SUBNETVAR}.4.4    ${HOSTPREFIX}-r4-n4.${DOMAIN} ${HOSTPREFIX}-r4-n4 cass-r4-n4
10.${SUBNETVAR}.4.5    ${HOSTPREFIX}-r4-n5.${DOMAIN} ${HOSTPREFIX}-r4-n5 cass-r4-n5
10.${SUBNETVAR}.4.6    ${HOSTPREFIX}-r4-n6.${DOMAIN} ${HOSTPREFIX}-r4-n6 cass-r4-n6
10.${SUBNETVAR}.4.7    ${HOSTPREFIX}-r4-n7.${DOMAIN} ${HOSTPREFIX}-r4-n7 cass-r4-n7
10.${SUBNETVAR}.4.8    ${HOSTPREFIX}-r4-n8.${DOMAIN} ${HOSTPREFIX}-r4-n8 cass-r4-n8
10.${SUBNETVAR}.4.9    ${HOSTPREFIX}-r4-n9.${DOMAIN} ${HOSTPREFIX}-r4-n9 cass-r4-n9
10.${SUBNETVAR}.4.10   ${HOSTPREFIX}-r4-n10.${DOMAIN} ${HOSTPREFIX}-r4-n10 cass-r4-n10

10.${SUBNETVAR}.0.101  driver1-${SUBNETVAR}.${DOMAIN} driver1-${SUBNETVAR} driver1.${DOMAIN} driver1
10.${SUBNETVAR}.0.102  driver2-${SUBNETVAR}.${DOMAIN} driver2-${SUBNETVAR} driver2.${DOMAIN} driver2
10.${SUBNETVAR}.0.103  driver3-${SUBNETVAR}.${DOMAIN} driver3-${SUBNETVAR} driver3.${DOMAIN} driver3
10.${SUBNETVAR}.0.104  driver4-${SUBNETVAR}.${DOMAIN} driver4-${SUBNETVAR} driver4.${DOMAIN} driver4
10.${SUBNETVAR}.0.105  driver5-${SUBNETVAR}.${DOMAIN} driver5-${SUBNETVAR} driver5.${DOMAIN} driver5
10.${SUBNETVAR}.0.106  driver6-${SUBNETVAR}.${DOMAIN} driver6-${SUBNETVAR} driver6.${DOMAIN} driver6
10.${SUBNETVAR}.0.107  driver7-${SUBNETVAR}.${DOMAIN} driver7-${SUBNETVAR} driver7.${DOMAIN} driver7
10.${SUBNETVAR}.0.108  driver8-${SUBNETVAR}.${DOMAIN} driver8-${SUBNETVAR} driver8.${DOMAIN} driver8
10.${SUBNETVAR}.0.109  driver9-${SUBNETVAR}.${DOMAIN} driver9-${SUBNETVAR} driver9.${DOMAIN} driver9
10.${SUBNETVAR}.0.110  driver10-${SUBNETVAR}.${DOMAIN} driver10-${SUBNETVAR} driver10.${DOMAIN} driver10
```

Read the report at http://facts.pt/kk9hwux ▶

This project was commissioned by Dell EMC.

Principled Technologies®

Facts matter.®