Real Application Clusters (RAC) in containers on IBM System Z

Implementing Oracle RAC with Podman

Table of contents

Introduction

This paper provides documentation in the form of examples explaining how to implement Podman with Oracle Real Application Clusters (RAC) on IBM System Z.

Terminology

- Podman host - The host operating system where Podman is run and where containers are built.
- RAC host - The host operating system running in the containers built on the Podman hosts.
- LPAR - A logical partition on an IBM System Z server.

Goal

Create a four-node RAC cluster running in Podman Containers on IBM System Z hardware that includes the following settings:

- The entire environment except disk will be running on a single IBM physical server
- Two logical partitions (LPARs) will be used to run the Red Hat 8.6 OS; the "Podman hosts"
- Each Podman host will support two RAC nodes which run in containers.
- One Podman host will support a client system for testing.
- All disk storage is 3390-M54 direct access storage devices (DASDs) located on an IBM storage area network (SAN) device.
- The RAC cluster will utilize one public network and two private networks.
- Oracle Grid Infrastructure version 19.16 to be installed.
- Oracle Relational Database Management System (RDBMS) version 19.16 to be installed.
- A sample CDB/PDB database to be created and tested.

Architecture diagram

Figure 1 depicts the typical architecture of a four-node RAC cluster deployed on two container hosts with one client node on an IBM LinuxOne server.

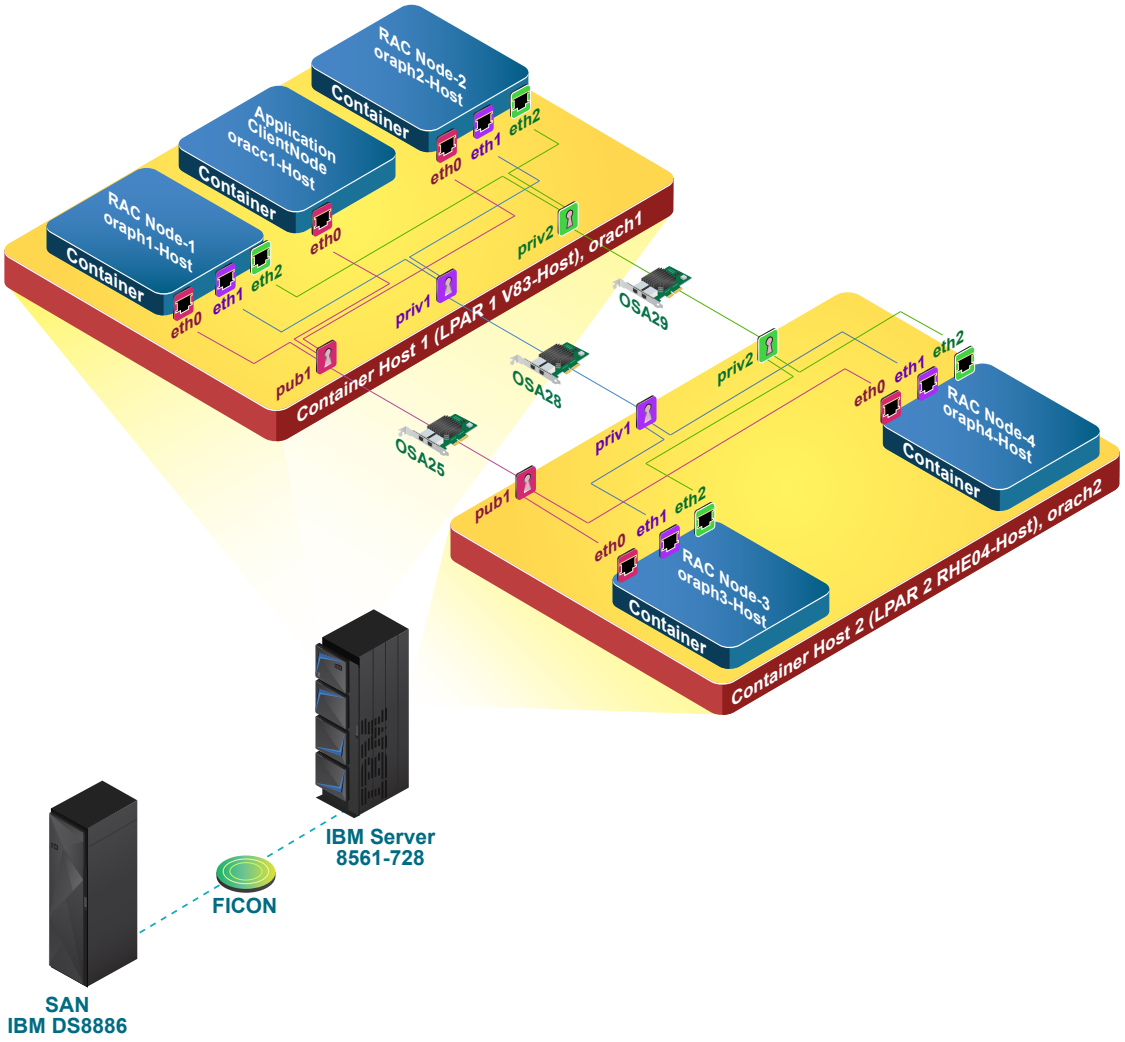

Figure 1. Architecture overview

Prerequisites

This section describes the software and network requirements that must be met before creating Podman images or containers.

Podman host software requirements

Podman host software requirements include:

- Red Hat 8.6 OS
- An active subscription to Red Hat Base and Appstream repositories by Podman hosts
    - The Red Hat provided Universal Base Image (UBI) repositories do not include RPMs required by OracleGrid Infrastructure (GI) or RDBMS
- Podman 4.0.2 or later

- Packages (Anaconda specified) needed:
    - graphical-server-environment
    - development
    - file-server
    - ftp-server
    - graphical-admin-tools
    - legacy-unix
    - network-file-system-client
    - remote-system-management

Podman host disk requirements

The Podman host disk requirements include:

- 40 GB for root
- 40 GB for swap
- 80 GB for container storage
- 40 GB for image build and staged Oracle code
- 80 GB for Oracle GI/RDBMS code areas

Note:

- 280 GB per Podman host is required.
- Datafiles and rollback areas are not included on shared disk.

Podman host network requirements

Podman host network requirements include:

- One network interface card (NIC) connected to a *public* network with access to Domain Name System (DNS)
- Two NICs connected to *private* networks to be used by RAC hosts for Highly Available Internet Protocol (HAIPs)

RAC host software requirements

RAC host software requirements include:

- Red Hat 8.6 OS (same OS as the Podman Host)
- Oracle Grid Infrastructure Release 19c (19.3), updated with Release Update (RU) 19.16, or later release updates
- Oracle Database Release 19c (19.3) updated with RU 19.16 or later release updates
- Other Oracle required RPMs

Client host software requirements

Client host software requirement include:

- Oracle Database 19c RDBMS 19.3 or later client

Test environment

The section describes the test environment and note that the plan is to create the entire Podman and RAC cluster on one System Z server.

Server hardware

- 8561-728 (z15) Serial: 36F98
- 40 TB memory
- 145 CPUs

SAN hardware

- IBM System Storage DS8886 (2831-981)
- 36 3390-54 (40 GB each) DASD devices

Logical partitions

LPAR1:

- Name "V83"
- Four shared CPUs
- 1.25 TB memory
- Three shared 10 Gbps OSAD (OSD_10GIG) cards
    - Addr: 1450 - NIC: pub1
    - Addr: 1480 - NIC: priv1
    - Addr: 1490 - NIC: priv2
- One 3390-54 DASD Addr: b800 (40 GB) for root
- One 3390-54 DASD Addr: b801 (40 GB) for swap
- One 3390-54 DASD Addr: b802 (40 GB) for Oracle code on RAC host oraph1
- One 3390-54 DASD Addr: b803 (40 GB) for Oracle code on RAC host oraph2
- Six 3390-54 DASD Addr: b804-b809 (240 GB) for scratch

LPAR2:

- Name "RH04"
- Four shared CPUs
- 1.25 TB memory
- Three shared 10 Gbps OSAD (OSD_10GIG) cards
    - Addr: 1453 - NIC: pub1
    - Addr: 1483 - NIC: priv1
    - Addr: 1493 - NIC: priv2
- One 3390-54 DASD Addr: b80b (40 GB) for root
- One 3390-54 DASD Addr: b80c (40 GB) for swap
- One 3390-54 DASD Addr: b80d (40 GB) for Oracle code on RAC host oraph3
- One 3390-54 DASD Addr: b80e (40 GB) for Oracle code on RAC host oraph4
- Six 3390-54 DASD Addr: b80f-b814 (240 GB) for scratch

Disk storage mapping

Disk storage used by RAC hosts is provided by Podman hosts. Key Podman disk storage areas with their mapping (if mapped) to RAC hosts is shown in the form Podman host " <->" RAC host.

Details are provided in later sections for these examples. Users can define names that meet their own requirements.

Container storage (not mapped to Podman):
Set in the `graphroot` parameter in /etc/containers/storage.conf

orach1:/scratch/containers/storage

orach2:/scratch/containers/storage

Note: This is displayed in Podman info command output.

The context directory (not mapped to Podman):
Used for Containerfile and other files used with Podman builds.
orach1:/scratch/image
orach2:/scratch/image

The Mapped directories are as follows:

Staging area for software used on RAC hosts:
orach1:/scratch/software/stage <-> oraph1:/software/stage (RW)
orach1:/scratch/software/stage <-> oraph2:/software/stage (RW)
orach2:/scratch/software/stage <-> oraph3:/software/stage (RW)
orach2:/scratch/software/stage <-> oraph4:/software/stage (RW)
orach1:/scratch/software/stage <-> oracc1:/software/stage (RW)

Oracle (base,grid/rdbms home, oraInventory):
orach1:/oraph1 <-> oraph1:/u01 (RW)
orach1:/oraph2 <-> oraph2:/u01 (RW)
orach2:/oraph3 <-> oraph3:/u01 (RW)
orach2:/oraph4 <-> oraph4:/u01 (RW)
orach1:/scratch/oracc1 <-> oracc1:/u01 (RW)

Needed for boot process:
orach1:/boot <-> oraph1:/boot (RO)
orach1:/boot <-> oraph2:/boot (RO)
orach2:/boot <-> oraph3:/boot (RO)
orach2:/boot <-> oraph4:/boot (RO)
orach1:/boot <-> oracc1:/boot (RO)

Time zone settings:
orach1:/etc/localtime <-> oraph1:/etc/localtime (RO)
orach1:/etc/localtime <-> oraph2:/etc/localtime (RO)
orach2:/etc/localtime <-> oraph3:/etc/localtime (RO)
orach2:/etc/localtime <-> oraph4:/etc/localtime (RO)
orach1:/etc/localtime <-> oracc1:/etc/localtime (RO)

Kernel file for allocating pages > 4KiB:
orach1:/dev/hugepages <-> oraph1:/dev/hugepages (RO)
orach1:/dev/hugepages <-> oraph2:/dev/hugepages (RO)
orach2:/dev/hugepages <-> oraph3:/dev/hugepages (RO)
orach2:/dev/hugepages <-> oraph4:/dev/hugepages (RO)
orach1:/dev/hugepages <-> oracc1:/dev/hugepages (RO)

Shared Disk (16 devices):
orach1:/dev/oracleasm/asm(x) <-> oraph1:/dev/asm-(x) (RW)
orach1:/dev/oracleasm/asm(x) <-> oraph2:/dev/asm-(x) (RW)
orach1:/dev/oracleasm/asm(x) <-> oraph3:/dev/asm-(x) (RW)
orach1:/dev/oracleasm/asm(x) <-> oraph4:/dev/asm-(x) (RW)
where: x in (b816, b817,...b825)

Network configuration

In this example, the test team created three RAC host 'macvlan' networks that map to the three Podman host networks. That is, the 'parent' of each ‘macvlan’ network is the associated Podman host NIC.

So, then you have:

Network	Pub/Priv	PNIC	RNIC	MTU	Mask	OSA/enet
129.40.1.0/27	Public	pub1	eth0	1500	255.255.255.224	25
12.1.1.0/24	Private	priv1	eth1	8992	255.255.255.0	28
13.1.1.0/24	Private	priv2	eth2	8992	255.255.255.0	29

Where:

PNIC is the network interface name as seen on the Podman hosts
RNIC is the network interface name as seen on the RAC hosts
OSA is the name of the Ethernet network device used

Hostnames and IP addresses used:

Hostname	IP	Use	Level	Desc
orach1	129.40.1.1	Public	Podman	Podman #1 hostname
orach1-I	12.1.1.101	Private	Podman	Podman #1 NIC1 hostname
orach1-i-	213.1.1.101	Private	Podman	Podman #1 NIC2 hostname
oraph1	129.40.1.11	Public	RAC	RAC #1 hostname
oraph1v	129.40.1.5	Public	RAC	RAC #1 VIP
oraph1-i	12.1.1.201	Private	RAC	RAC #1 NIC1 hostname
oraph1-i-2	13.1.1.201	Private	RAC	RAC #1 NIC2 hostname
oraph2	129.40.1.12	Public	RAC	RAC #2 hostname
oraph2v	129.40.1.6	Public	RAC	RAC #2 VIP
oraph2-i	12.1.1.202	Private	RAC	RAC #2 NIC1 hostname
oraph2-i-2	13.1.1.202	Private	RAC	RAC #2 NIC2 hostname
#				
orach2	129.40.1.2	Public	Podman	Podman #2 hostname
orach2-i	12.1.1.102	Private	Podman	Podman #1 NIC1 hostname
orach2-i-2	13.1.1.102	Private	Podman	Podman #1 NIC2 hostname
oraph3	129.40.1.13	Public	RAC	RAC #3 hostname
oraph3v	129.40.1.7	Public	RAC	RAC #3 VIP
oraph3-i	12.1.1.203	Private	RAC	RAC #3 NIC1 hostname
oraph3-i-2	13.1.1.203	Private	RAC	RAC #3 NIC2 hostname
oraph4	129.40.1.14	Public	RAC	RAC #4 hostname
oraph4v	129.40.1.8	Public	RAC	RAC #4 VIP
oraph4-i	12.1.1.204	Private	RAC	RAC #4 NIC1 hostname
oraph4-i-2	13.1.1.204	Private	RAC	RAC #4 NIC2 hostname
#				
orascanpm14	129.40.1.20	Public	RAC	SCAN hostname for RAC cluster
orascanpm14	129.40.1.21	Public	RAC	SCAN hostname for RAC cluster
orascanpm14	129.40.1.22	Public	RAC	SCAN hostname for RAC cluster
#				
oracc1	129.40.1.24	Public	RAC	client hostname/IP
oracc2	129.40.1.25	Public	RAC	Spare hostname/IP - not used

Additional IP addresses:

DNS Server: 129.40.106.1
Def Gateway: 129.40.1.30
Broadcast: 129.40.1.31

Oracle installation areas

On RAC hosts, critical Oracle files are at:

/u01/base	- Oracle base for GI/RDBMS
/u01/grid	- GI home
/u01/db	- RDBMS home
/u01/oraInventory	- Oracle inventory

On client host, critical Oracle files are at:

/u01/base	- Oracle base for RDBMS
/u01/client	- RDBMS home
/u01/oraInventory	- Oracle inventory

Podman details

This section describes the characteristics of the installed Podman.

```
[root@orach1 ~]# Podman info
host:
  arch: s390x
  buildahVersion: 1.26.2
  cgroupControllers:
  - cpuset
  - cpu
  - cpuacct
  - blkio
  - memory
  - devices
  - freezer
  - net_cls
  - perf_event
  - net_prio
  - hugetlb
  - pids
  - rdma
  cgroupManager: systemd
  cgroupVersion: v1
  conmon:
    package: conmon-2.1.2-2.module+el8.6.0+15917+093ca6f8.s390x
    path: /usr/bin/conmon
    version: 'conmon version 2.1.2, commit: cb2793b54ccb07ad200e41d307a592a6ee3ccc14'
  cpuUtilization:
    idlePercent: 0
    systemPercent: 100
    userPercent: 0
  cpus: 8
  distribution:
    distribution: '"rhel"'
    version: "8.6"
  eventLogger: file
  hostname: orach1
  idMappings:
    gidmap: null
    uidmap: null
```

```
  kernel: 4.18.0-372.26.1.el8_6.s390x
  linkmode: dynamic
  logDriver: k8s-file
  memFree: 1333636694016
  memTotal: 1352823521280
  networkBackend: cni
  ociRuntime:
    name: runc
    package: runc-1.1.3-2.module+el8.6.0+15917+093ca6f8.s390x
    path: /usr/bin/runc
    version: |-
      runc version 1.1.3
      spec: 1.0.2-dev
      go: go1.17.7
      libseccomp: 2.5.2
  os: linux
  remoteSocket:
    path: /run/podman/podman.sock
  security:
    apparmorEnabled: false
    capabilities:
CAP_NET_RAW,CAP_CHOWN,CAP_DAC_OVERRIDE,CAP_FOWNER,CAP_FSETID,CAP_KILL,CAP_NET_BIND_SERVICE,CAP_SETF
CAP,CAP_SETGID,CAP_SETPCAP,CAP_SETUID,CAP_SYS_CHROOT
    rootless: false
    seccompEnabled: true
    seccompProfilePath: /usr/share/containers/seccomp.json
    selinuxEnabled: false
  serviceIsRemote: false
  slirp4netns:
    executable: /usr/bin/slirp4netns
    package: slirp4netns-1.2.0-2.module+el8.6.0+15917+093ca6f8.s390x
    version: |-
      slirp4netns version 1.2.0
      commit: 656041d45cfca7a4176f6b7eed9e4fe6c11e8383
      libslirp: 4.4.0
      SLIRP_CONFIG_VERSION_MAX: 3
      libseccomp: 2.5.2
  swapFree: 44311900160
  swapTotal: 44311900160
  uptime: 9h 31m 21.39s (Approximately 0.38 days)
plugins:
  log:
  - k8s-file
  - none
  - passthrough
  - journald
  network:
  - bridge
  - macvlan
  - ipvlan
  volume:
  - local
registries:
  search:
  - registry.access.redhat.com
  - registry.redhat.io
  - docker.io
store:
  configFile: /etc/containers/storage.conf
```

```
  containerStore:
    number: 0
    paused: 0
    running: 0
    stopped: 0
  graphDriverName: overlay
  graphOptions:
    overlay.mountopt: nodev,metacopy=on
  graphRoot: /scratch/containers/storage
  graphRootAllocated: 252580130816
  graphRootUsed: 63021056
  graphStatus:
    Backing Filesystem: extfs
    Native Overlay Diff: "false"
    Supports d_type: "true"
    Using metacopy: "true"
  imageCopyTmpDir: /var/tmp
  imageStore:
    number: 0
  runRoot: /run/containers/storage
  volumePath: /scratch/containers/storage/volumes
version:
  APIVersion: 4.1.1
  Built: 1657551385
  BuiltTime: Mon Jul 11 09:56:25 2022
  GitCommit: ""
  GoVersion: go1.17.7
  Os: linux
  OsArch: linux/s390x
  Version: 4.1.1
```

Configure Podman hosts

The Podman (or container) hosts must be configured appropriately to support multiple containers required for Oracle RAC nodes.

Set kernel parms

On both Podman hosts, in /etc/sysctl.conf:

```
fs.aio-max-nr=1048576
fs.file-max = 6815744
net.core.rmem_max = 4194304
net.core.rmem_default = 262144
net.core.wmem_max = 1048576
net.core.wmem_default = 262144
vm.nr_hugepages=16384
```

Create file systems for the Oracle code

On Podman host orach1, create file systems for RAC hosts oraph1 and oraph2.
On Podman host orach2, create file systems for RAC hosts oraph3 and oraph4.
These Podman host file systems are mapped to RAC hosts in directory, ‘/u01’ and will be used for Oracle installation areas.

On both Podman hosts:

```
vgcreate <RAC_host>_vg <disk>
lvcreate -L 40G -n <RAC_host>_lv <RAC_host>_vg
mkfs -t ext4 /dev/<RAC_host>_vg/<RAC_host>_lv
```

Update /etc/fstab:
```
/dev/<RAC_host>_vg/<RAC_host>_lv /<RAC_host> ext4 defaults 0 2
```

Create mount point:
```
mkdir /<RAC_host>
```

Mount:
```
mount /<RAC_host>
```

Check:
```
df -h /<RAC_host>
```

where:
- RAC_host is in oraph<1,2,3,4>
- Disk is a full volume DASD partition in /dev/disk/by-path

For example, with Podman host orach2, RAC host oraph3:

```
[root@orach2 ~]# vgcreate oraph3_vg /dev/disk/by-path/ccw-0.0.b80d-part1
[root@orach2 ~]# lvcreate -L 40G -n oraph3_lv oraph3_vg
[root@orach2 ~]# mkfs -t ext4 /dev/oraph3_vg/oraph3_lv
```

Update /etc/fstab:
```
/dev/oraph3_vg/oraph3_lv /oraph3 ext4 defaults 0 2
```

Create mount point:
```
[root@orach2 ~]# mkdir /oraph3
```

Mount:
```
[root@orach2 ~]# mount /oraph3
```

Check:
```
[root@orach2 ~]# df -h /oraph3
Filesystem                      Size  Used Avail Use% Mounted on
/dev/mapper/oraph3_vg-oraph3_lv   40G   19G   19G  51% /oraph3
```

Client RAC host oracc1 will use this directory on orach1 for Oracle code:

```
mkdir -p /scratch/oracc1
```

Check shared memory file system mount

On both Podman hosts:

```
[root@orach1 ~]# df -h /dev/shm
       Filesystem      Size  Used Avail Use% Mounted on
       tmpfs            50G   84K   50G   1% /dev/shm
```

Check chronyd

On both Podman hosts:

```
[root@orach1 ~]# systemctl status chronyd.service
chronyd.service - NTP client/server
   Loaded: loaded (/usr/lib/systemd/system/chronyd.service; enabled; vendor pre>
   Active: active (running) since Tue 2022-10-25 01:34:44 CDT; 9h ago
     Docs: man:chronyd(8)
           man:chrony.conf(5)
...
```

Enable real-time mode for Oracle RAC processes

On both Podman hosts:

Create /etc/systemd/system/podman-rac-cgroup.service:

```
[Unit]
Description=Populate Cgroups with real time chunk on machine restart
After=multi-user.target
[Service]
Type=oneshot
ExecStart=/bin/bash -c "echo 950000 > /sys/fs/cgroup/cpu,cpuacct/machine.slice/cpu.rt_runtime_us &&
\
/bin/systemctl restart podman-restart.service"
StandardOutput=journal
CPUAccounting=yes
Slice=machine.slice
[Install]
WantedBy=multi-user.target
```

Create and start the `oneshot systemd` service:

```
[root@orach2 system]# systemctl daemon-reload

[root@orach2 system]# systemctl enable podman-restart.service
Created symlink /etc/systemd/system/default.target.wants/podman-restart.service →
/usr/lib/systemd/system/podman-restart.service.

[root@orach2 system]# systemctl enable podman-rac-cgroup.service --now
Created symlink /etc/systemd/system/multi-user.target.wants/podman-rac-cgroup.service →
/etc/systemd/system/podman-rac-cgroup.service.
```

Check the status of the `oneshot systemd` service:

```
[root@orach2 system]# systemctl status podman-restart.service
podman-restart.service - Podman Start All Containers With Restart Policy Set To Always
   Loaded: loaded (/usr/lib/systemd/system/podman-restart.service; enabled; vendor preset:
disabled)
   Active: active (exited) since Tue 2022-10-25 11:10:57 CDT; 18s ago
     Docs: man:podman-start(1)
  Process: 60487 ExecStart=/usr/bin/podman $LOGGING start --all --filter restart-policy=always
(code=exited, status=0/SUCCESS)
 Main PID: 60487 (code=exited, status=0/SUCCESS)
Oct 25 11:10:57 orach2 systemd[1]: Starting Podman Start All Containers With Restart Policy Set To
Always...
```

```
Oct 25 11:10:57 orach2 podman[60487]: time="2022-10-25T11:10:57-05:00" level=info
msg="/usr/bin/podman filtering at log level info"
Oct 25 11:10:57 orach2 podman[60487]: time="2022-10-25T11:10:57-05:00" level=info msg="Not using
native diff for overlay, this may cause degraded performance for building images: kernel ha>
Oct 25 11:10:57 orach2 podman[60487]: time="2022-10-25T11:10:57-05:00" level=info msg="Setting
parallel job count to 25"
Oct 25 11:10:57 orach2 systemd[1]: Started Podman Start All Containers With Restart Policy Set To
Always.
```

Configure shared device persistence

The Podman host to RAC host device mapping is:

```
/dev/oracleasm/asmb8${i} <-> /dev/asm-b8{i}
```

where i in <16,17,18,19,1a,1b,1c,1d,1e,20,21,22,23,24,25>

Devices need to be referred to by their full path which includes to DASD device number.
On both Podman hosts, all shared devices should be specified in /etc/udev/rules.d/99-udev-oracle.rules on both Podman hosts:

```
[root@orach1 ~]# cat /etc/udev/rules.d/99-udev-oracle.rules

ACTION=="add|change", ENV{ID_PATH}=="ccw-0.0.b816", SYMLINK+="oracleasm/asmb816"
ACTION=="add|change", ENV{ID_PATH}=="ccw-0.0.b817", SYMLINK+="oracleasm/asmb817"
ACTION=="add|change", ENV{ID_PATH}=="ccw-0.0.b818", SYMLINK+="oracleasm/asmb818"
...
ACTION=="add|change", ENV{ID_PATH}=="ccw-0.0.b823", SYMLINK+="oracleasm/asmb823"
ACTION=="add|change", ENV{ID_PATH}=="ccw-0.0.b824", SYMLINK+="oracleasm/asmb824"
ACTION=="add|change", ENV{ID_PATH}=="ccw-0.0.b825", SYMLINK+="oracleasm/asmb825"
```

On both Podman hosts, make UDEV aware:

```
[root@orach1 ~]# udevadm control -R
[root@orach1 ~]# udevadm trigger
```

On both Podman hosts, check /dev/oracleasm/asm* to verify symlinks:

```
[root@orach1 ~]# ls -al /dev/oracleasm/asm*
lrwxrwxrwx 1 root root 9 Oct 25 15:18 /dev/oracleasm/asmb816 -> ../dasdk1
lrwxrwxrwx 1 root root 9 Oct 25 15:18 /dev/oracleasm/asmb817 -> ../dasdl1
lrwxrwxrwx 1 root root 9 Oct 25 15:18 /dev/oracleasm/asmb818 -> ../dasdm1
...
lrwxrwxrwx 1 root root 9 Oct 25 15:18 /dev/oracleasm/asmb823 -> ../dasdx1
lrwxrwxrwx 1 root root 9 Oct 25 15:18 /dev/oracleasm/asmb824 -> ../dasdy1
lrwxrwxrwx 1 root root 9 Oct 25 15:18 /dev/oracleasm/asmb825 -> ../dasdz1
```

Initialize shared ASM DASD

On just one Podman host, initialize the shared devices:

```
[root@orach1 ~]# for i in 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25
do
   dd if=/dev/zero of=/dev/oracleasm/asmb8${i} bs=1024k count=1024
done
```

Add additional Red Hat Package Managers (RPMs) to support configuration

On both Podman hosts, install the RPMs mentioned in this section using the yum tool as required. The following RPM is needed if you want to run ifdown and ifup commands:

- NetworkManager-initscripts-updown-1.36.0-7.el8_6.noarch

The following RPMs are needed to run the TigerVNC server on Podman hosts:

- fltk-1.3.4-5.el8.s390x
- mesa-libGLU-9.0.0-15.el8.s390x
- tigervnc-1.12.0-4.el8.s390x
- tigervnc-icons-1.12.0-4.el8.noarch
- tigervnc-license-1.12.0-4.el8.noarch
- tigervnc-selinux-1.12.0-4.el8.noarch
- tigervnc-server-1.12.0-4.el8.s390x
- tigervnc-server-minimal-1.12.0-4.el8.s390x

Create Podman images and containers

Except as noted, run all the steps specified in this section on both Podman hosts.

Create a staging area for software

```
[root@orach1 ~]# mkdir -p /scratch/software/stage
[root@orach1 ~]# chmod a+rwx /scratch/software/stage
```

Acquire the following Oracle software (at the time of this paper, version 19.16 is the latest, and so verify if that is still the case or get the latest version) from Oracle and place them in the stage area (/scratch/software/stage) on orach1 only:

Oracle Database 19c Grid Infrastructure 19.3
LINUX.ZSERIES64_193000_grid_home.zip

Oracle Database 19c RDBMS 19.3
LINUX.ZSERIES64_193000_db_home.zip

Oracle Database 19c RDBMS 19.3 client
LINUX.ZSERIES64_193000_client.zip

Oracle GI and RDBMS maintenance 34130714
p34130714_190000_Linux-zSer.zip

OPatch v12.2.0.1.32
p6880880_180000_Linux-zSer.zip

OJVM release update 19.16
p34086870_190000_Linux-zSer.zip

Create the Podman image build directory

Create scripts and files used to build and configure RAC hosts.

```
[root@orach1 ~]# mkdir /scratch/image
[root@orach1 ~]# cd /scratch/image
```

Create RAC host /etc/resolv.conf

Create /scratch/image/resolv.conf for RAC hosts. This gets copied during the container startup time to /etc/resolv.conf on the RAC host.

```
search pbm.ihost.com
nameserver 129.40.106.1
```

Create RAC host hostfile

Create /scratch/image/hostfile for container hosts. This gets copied during the container startup time to /etc/hosts on the RAC host.

```
#
# /etc/hosts for Oracle certifications
#
# Localhost
127.0.0.1       localhost

# special IPv6 addresses
::1             localhost ipv6-localhost ipv6-loopback
fe00::0         ipv6-localnet
ff00::0         ipv6-mcastprefix
ff02::1         ipv6-allnodes
ff02::2         ipv6-allrouters
ff02::3         ipv6-allhosts

# Public
129.40.1.11     oraph1.pbm.ihost.com oraph1
129.40.1.12     oraph2.pbm.ihost.com oraph2
129.40.1.13     oraph3.pbm.ihost.com oraph3
129.40.1.14     oraph4.pbm.ihost.com oraph4
129.40.1.24     oracc1.pbm.ihost.com oracc1
129.40.1.25     oracc2.pbm.ihost.com oracc2

# Private NIC #1
12.1.1.201      oraph1-i.pbm.ihost.com oraph1-i
12.1.1.202      oraph2-i.pbm.ihost.com oraph2-i
12.1.1.203      oraph3-i.pbm.ihost.com oraph3-i
12.1.1.204      oraph4-i.pbm.ihost.com oraph4-i

# Private NIC #2
13.1.1.201      oraph1-i-2.pbm.ihost.com oraph1-i-2
13.1.1.202      oraph2-i-2.pbm.ihost.com oraph2-i-2
13.1.1.203      oraph3-i-2.pbm.ihost.com oraph3-i-2
13.1.1.204      oraph4-i-2.pbm.ihost.com oraph4-i-2

# Virtual Internet Protocol (VIPs)
129.40.1.5      oraph1v.pbm.ihost.com oraph1v
129.40.1.6      oraph2v.pbm.ihost.com oraph2v
129.40.1.7      oraph3v.pbm.ihost.com oraph3v
```

```
129.40.1.8      oraph4v.pbm.ihost.com oraph4v

# Oracle SCAN
# 129.40.1.20   orascanpm14.pbm.ihost.com orascanpm14
# 129.40.1.21   orascanpm14.pbm.ihost.com orascanpm14
# 129.40.1.22   orascanpm14.pbm.ihost.com orascanpm14
```

Create RAC host limits.conf

Create /scratch/image/limits.conf to be used by RAC hosts.

```
#
# Oracle recommended values
#
# memlock should be 90% of RAM in kb
#   Total memory=1321116720 kB
#   90% memory=1,321,116,720*.9=1,189,005,048
#
oracle  soft    nofile  1024
oracle  hard    nofile  65536
oracle  soft    nproc   2047
oracle  hard    nproc   16384
oracle  soft    stack   10240
oracle  hard    stack   10240
oracle  soft    memlock 1189005048
oracle  hard    memlock 1189005048
oracle  soft    core    unlimited
oracle  hard    core    unlimited
```

Create the script to perform startup config

Create script /scratch/image/setupContainerEnv.sh to:

- Map Podman host shared disks to RAC hosts
- Set up networking
- Get the file later moved to /etc/rc.local on RAC hosts

```
#!/bin/bash
#
# Set owner/permissions for shared devices (RAC hosts only)
#
HN=`hostname`
if [ "$HN" != "oracc1" ]
then
   for i in 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25
   do
      chown oracle:asmadmin /dev/asm-b8${i}
      chmod 660 /dev/asm-b8${i}
   done
fi
#
# Fix Podman host network routing
#
ip route del default via 12.1.1.1
ip route del default via 13.1.1.1
ip route del default via 129.40.181.126
ip route add default via 129.40.181.126
```

```
#
# Make sure DNS information is correct
#
cat /opt/scripts/startup/resolv.conf > /etc/resolv.conf
#
# Put in correct /etc/hosts
#
cat /opt/scripts/startup/hostfile > /etc/hosts
#
# Set memlock to 90% of memory limit
#
OFILE=/etc/security/limits.conf
CONTAINER_MEMORY=$(cat /sys/fs/cgroup/memory/memory.limit_in_bytes)
NEW_MEMLOCK=$((($CONTAINER_MEMORY/1024)*9/10))
sed -i "/memlock/d" $OFILE
echo "oracle  soft    memlock ${NEW_MEMLOCK}" >> $OFILE
echo "oracle  hard    memlock ${NEW_MEMLOCK}" >> $OFILE
#
# Reset all units that have failed status
#
systemctl reset-failed
```

Create Containerfile

Create /scratch/image/Containerfile

```
#
# Get RHEL8.6
FROM registry.access.redhat.com/ubi8/ubi:8.6-943
#
# Install Oracle required RPMs:
RUN yum install -y bc binutils elfutils-libelf elfutils-libelf-devel fontconfig gcc gcc-c++ glibc
glibc-devel initscripts ksh libaio libaio-devel libattr-devel libgcc libgfortran libibverbs libnsl
libnsl2 librdmacm libstdc++ libstdc++-devel libX11 libXau libXaw libxcb libXi libXrender libXtst
make net-tools pam pam-devel policycoreutils policycoreutils-python-utils smartmontools sysstat
#
# Install s390 specific utility packages
RUN yum install -y s390utils
#
# Install X packages to Work with X11 forward in SSH
RUN yum install -y xorg-x11-xauth xorg-x11-fonts-* xorg-x11-utils xterm dbus-x11
#
# Install other RPMs I know we will need
RUN yum install -y glibc-langpack-en hostname bind-utils iproute iputils java-1.8.0-openjdk-
headless javapackages-filesystem kernel-devel openssh openssh-clients openssl-pkcs11 openssh-server
unzip zip
#
# Cleanup temp files used by yum
RUN yum clean all
#
##
#
# Environment variables required for this build (do NOT change)
# -------------------------------------------------------------
# Environment Variables
ENV container=true \
    SCRIPT_DIR=/opt/scripts/startup \
    RESOLVCONFENV="resolv.conf" \
```

```
    HOSTFILEENV="hostfile" \
    LIMITS_CONF="limits.conf" \
    SETUPCONTAINERENV="setupContainerEnv.sh"
#
# Copy HOST files to new OS
COPY  $SETUPCONTAINERENV $LIMITS_CONF $HOSTFILEENV $RESOLVCONFENV $SCRIPT_DIR/
#
# Setup users/groups
RUN groupadd -g 54321 oinstall && \
 groupadd -g 54322 dba && \
 groupadd -g 54327 asmdba && \
 groupadd -g 54328 asmoper && \
 groupadd -g 54329 asmadmin && \
 useradd -u 54321 -g oinstall -G oinstall,asmadmin,asmdba,asmoper,dba oracle
#
# Some additional config
RUN rm -f /usr/lib/systemd/system/dnf-makecache.service && \
 echo "$SCRIPT_DIR/$SETUPCONTAINERENV" >> /etc/rc.local && \
 chmod +x /etc/rc.d/rc.local && \
 echo "NOZEROCONF=yes" >> /etc/sysconfig/network && \
 echo "net.ipv4.conf.eth1.rp_filter=2" >> /etc/sysctl.conf && \
 echo "net.ipv4.conf.eth2.rp_filter=2" >> /etc/sysctl.conf && \
 touch /etc/ntp.conf && \
 echo 'LANG="en_US.UTF-8"' >> /etc/locale.conf && \
 setcap 'cap_net_admin,cap_net_raw+ep' /usr/bin/ping && \
 cp "$SCRIPT_DIR/$LIMITS_CONF" /etc/security/ && \
 sync
#
USER root
WORKDIR /root
VOLUME ["/stage/software"]
VOLUME ["/u01"]
CMD ["/usr/sbin/init"]
# End of the Containerfile
```

Create the Oracle RAC on Podman image

On both Podman hosts, create an OS image:

```
[root@orach1 ~]# cd /scratch/image

[root@orach1 image]# export RH_version=8.6

[root@orach1 image]# export DB_version=19.16

[root@orach1 image]# podman build --force-rm=true --no-cache=true \
  -t oracle/database-rac:${RH_version}-${DB_version} -f Containerfile .
```

List Podman images:

```
[root@orach1 image]# podman images
REPOSITORY                              TAG         IMAGE ID      CREATED        SIZE
localhost/oracle/database-rac           8.6-19.16   9bec8b1f5d34  3 minutes ago  1.12 GB
registry.access.redhat.com/ubi8/ubi  8.6-943     e683387831b3  2 months ago   225 MB
```

Create RAC host networks

On both Podman hosts, create networks:

Note: For better Network performance, IBM recommends to set MTU=8992 for internal connectivity.

```
[root@orach1 image]# podman network create -d macvlan \
  --subnet=129.40.1.0/27 --gateway=129.40.1.30 \
  -o parent=pub1 -o mtu=1500 pm_pub1

[root@orach1 image]# podman network create -d macvlan \
  --subnet=12.1.1.0/24 \
  -o parent=priv1 -o mtu=8992 pm_priv1

[root@orach1 image]# podman network create -d macvlan \
  --subnet=13.1.1.0/24 \
  -o parent=priv2 -o mtu=8992 pm_priv2
```

List podman networks:

```
[root@orach1 image]# podman network ls
NETWORK ID    NAME        DRIVER
2d871d1285fe  pm_priv1    macvlan
7dd5d7263eb7  pm_priv2    macvlan
34da88cc1137  pm_pub1     macvlan
2f259bab93aa  podman      bridge
```

Note: When using macvlan, you will find that a Podman host will not be able to connect to its RAC hosts and the RAC hosts will not be able to connect their Podman hosts). This is a macvlan limitation. Check the USING DOCKER MACVLAN NETWORKS page that describes the issue and a work-around (that we don't need to do).

Create RAC host containers

Note:

GRID requirement - shmmax
- 50% of memory bytes
- Total memory = 64 GB
- Half memory = 32 GB
- Half memory = 34,359,738,368 bytes
- kernel.shmmax = 34359738368

GRID requirement - shmall
- Greater than or equal to shmmax in 4 KB blocks
- 34359738368/4096 = 8,388,608
- kernel.shmall = 8388608

On each Podman host, create the container:
- On orach1, create containers oraph1 and oraph2
- On orach2, create containers oraph3 and oraph4

Example for Podman host orach1, RAC host oraph1:

```
[root@orach1 image]# HN=oraph1
[root@orach1 image]# podman create -t -i \
  --hostname ${HN} \
```

```
  --volume /boot:/boot:ro \
  --volume /dev/hugepages:/dev/hugepages \
  --shm-size 12G \
  --dns-search=pbm.ihost.com \
  --dns=129.40.106.1 \
  --device=/dev/oracleasm/asmb816:/dev/asm-b816:rw  \
  --device=/dev/oracleasm/asmb817:/dev/asm-b817:rw  \
  --device=/dev/oracleasm/asmb818:/dev/asm-b818:rw  \
  --device=/dev/oracleasm/asmb819:/dev/asm-b819:rw  \
  --device=/dev/oracleasm/asmb81a:/dev/asm-b81a:rw  \
  --device=/dev/oracleasm/asmb81b:/dev/asm-b81b:rw  \
  --device=/dev/oracleasm/asmb81c:/dev/asm-b81c:rw  \
  --device=/dev/oracleasm/asmb81d:/dev/asm-b81d:rw  \
  --device=/dev/oracleasm/asmb81e:/dev/asm-b81e:rw  \
  --device=/dev/oracleasm/asmb81f:/dev/asm-b81f:rw  \
  --device=/dev/oracleasm/asmb820:/dev/asm-b820:rw  \
  --device=/dev/oracleasm/asmb821:/dev/asm-b821:rw  \
  --device=/dev/oracleasm/asmb822:/dev/asm-b822:rw  \
  --device=/dev/oracleasm/asmb823:/dev/asm-b823:rw  \
  --device=/dev/oracleasm/asmb824:/dev/asm-b824:rw  \
  --device=/dev/oracleasm/asmb825:/dev/asm-b825:rw  \
  --privileged=false  \
  --volume /scratch/software/stage:/software/stage \
  --volume /${HN}:/u01 \
  --volume /etc/localtime:/etc/localtime:ro \
  --cpuset-cpus 0-3 \
  --memory 64G \
  --memory-swap 96G \
  --sysctl 'kernel.sem=250 32000 100 128' \
  --sysctl 'kernel.shmall=8388608'  \
  --sysctl 'kernel.shmmax=34359738368'  \
  --sysctl 'kernel.shmmni=4096' \
  --sysctl 'net.ipv4.ip_local_port_range=9000 65500' \
  --sysctl 'net.ipv4.conf.eth1.rp_filter=2' \
  --sysctl 'net.ipv4.conf.eth2.rp_filter=2' \
  --cap-add=SYS_NICE \
  --cap-add=SYS_RESOURCE \
  --cap-add=NET_ADMIN \
  --cap-add=AUDIT_WRITE \
  --cap-add=AUDIT_CONTROL \
  --restart=always \
  --cpu-rt-runtime=95000 \
  --ulimit rtprio=99  \
  --systemd=true \
  --name ${HN} \
oracle/database-rac:8.6-19.16
```

Create the client container

Create the client container on Podman host orach1 only.

Note:

GRID requirement - shmmax
- 50% of memory bytes
- Total memory = 16GB
- Half memory = 8GB

Half memory = 8,589,934,592 bytes
kernel.shmmax = 8589934592

GRID requirement - shmall
Greater than or equal to shmmax in 4 KB blocks
8589934592/4096 = 2,097,152
kernel.shmall = 2097152

```
[root@orach1 image]# HN=oracc1
[root@orach1 image]# podman create -t -i \
  --hostname ${HN} \
  --volume /boot:/boot:ro \
  --volume /dev/hugepages:/dev/hugepages \
  --shm-size 12G \
  --dns-search=pbm.ihost.com \
  --dns=129.40.106.1 \
  --privileged=false  \
  --volume /scratch/software/stage:/software/stage \
  --volume /scratch/oracc1:/u01 \
  --volume /etc/localtime:/etc/localtime:ro \
  --cpuset-cpus 0-3 \
  --memory 16G \
  --memory-swap 24G \
  --sysctl 'kernel.sem=250 32000 100 128' \
  --sysctl 'kernel.shmall=2097152'  \
  --sysctl 'kernel.shmmax=8589934592'  \
  --sysctl 'kernel.shmmni=4096' \
  --sysctl 'net.ipv4.ip_local_port_range=9000 65500' \
  --cap-add=SYS_NICE \
  --cap-add=SYS_RESOURCE \
  --cap-add=NET_ADMIN \
  --cap-add=AUDIT_WRITE \
  --cap-add=AUDIT_CONTROL \
  --restart=always \
  --cpu-rt-runtime=95000 \
  --ulimit rtprio=99  \
  --systemd=true \
  --name ${HN} \
oracle/database-rac:8.6-19.16
```

Assign networks to the containers

On Podman host orach1:

```
[root@orach1 ~]# podman network disconnect podman oraph1
[root@orach1 ~]# podman network connect pm_pub1 --ip 129.40.1.11 oraph1
[root@orach1 ~]# podman network connect pm_priv1 --ip 12.1.1.201 oraph1
[root@orach1 ~]# podman network connect pm_priv2 --ip 13.1.1.201 oraph1
#
[root@orach1 ~]# podman network disconnect podman oraph2
[root@orach1 ~]# podman network connect pm_pub1 --ip 129.40.1.12 oraph2
[root@orach1 ~]# podman network connect pm_priv1 --ip 12.1.1.202 oraph2
[root@orach1 ~]# podman network connect pm_priv2 --ip 13.1.1.202 oraph2
#
[root@orach1 ~]# podman network disconnect podman oracc1
[root@orach1 ~]# podman network connect pm_pub1 --ip 129.40.1.24 oracc1
```

On Podman host orach2:

```
[root@orach2 ~]# podman network disconnect podman oraph3
[root@orach2 ~]# podman network connect pm_pub1 --ip 129.40.1.13 oraph3
[root@orach2 ~]# podman network connect pm_priv1 --ip 12.1.1.203 oraph3
[root@orach2 ~]# podman network connect pm_priv2 --ip 13.1.1.203 oraph3
#
[root@orach2 ~]# podman network disconnect podman oraph4
[root@orach2 ~]# podman network connect pm_pub1 --ip 129.40.1.14 oraph4
[root@orach2 ~]# podman network connect pm_priv1 --ip 12.1.1.204 oraph4
[root@orach2 ~]# podman network connect pm_priv2 --ip 13.1.1.204 oraph4
```

Start the containers

On Podman host orach1:

```
[root@orach1 ~]# podman start oraph1
[root@orach1 ~]# podman start oraph2
[root@orach1 ~]# podman start oracc1
```

On Podman host orach2:

```
[root@orach2 ~]# podman start oraph3
[root@orach2 ~]# podman start oraph4
```

Configure RAC hosts

After the containers are started, each RAC host needs initial configuration to be done before Oracle Grid Infrastructure can be installed.

Create paths and change permissions

For each Podman host/RAC host, you need to create paths/permissions.

Example for Podman host orach2, RAC host oraph3:

```
[root@orach2 ~]# HN=oraph3
[root@orach2 ~]# podman exec ${HN} /bin/bash -c "rm -fr /u01/base"
[root@orach2 ~]# podman exec ${HN} /bin/bash -c "rm -fr /u01/db"
[root@orach2 ~]# podman exec ${HN} /bin/bash -c "rm -fr /u01/grid"
[root@orach2 ~]# podman exec ${HN} /bin/bash -c "rm -fr /u01/oraInventory"
[root@orach2 ~]# podman exec ${HN} /bin/bash -c "mkdir -p /u01/base"
[root@orach2 ~]# podman exec ${HN} /bin/bash -c "mkdir -p /u01/db"
[root@orach2 ~]# podman exec ${HN} /bin/bash -c "mkdir -p /u01/grid"
[root@orach2 ~]# podman exec ${HN} /bin/bash -c "mkdir -p /u01/oraInventory"
[root@orach2 ~]# podman exec ${HN} /bin/bash -c "chown -R oracle:oinstall /u01/base"
[root@orach2 ~]# podman exec ${HN} /bin/bash -c "chown -R oracle:oinstall /u01/db"
[root@orach2 ~]# podman exec ${HN} /bin/bash -c "chown -R oracle:oinstall /u01/grid"
[root@orach2 ~]# podman exec ${HN} /bin/bash -c "chown -R oracle:oinstall /u01/oraInventory"
```

On Podman host orach1, create client directories:

```
[root@orach1 ~]# podman exec oracc1 /bin/bash -c "rm -fr /u01/base"
[root@orach1 ~]# podman exec oracc1 /bin/bash -c "rm -fr /u01/client"
[root@orach1 ~]# podman exec oracc1 /bin/bash -c "rm -fr /u01/oraInventory"
```

```
[root@orach1 ~]# podman exec oracc1 /bin/bash -c "mkdir -p /u01/base"
[root@orach1 ~]# podman exec oracc1 /bin/bash -c "mkdir -p /u01/client"
[root@orach1 ~]# podman exec oracc1 /bin/bash -c "mkdir -p /u01/oraInventory"
[root@orach1 ~]# podman exec oracc1 /bin/bash -c "chown -R oracle:oinstall /u01/base"
[root@orach1 ~]# podman exec oracc1 /bin/bash -c "chown -R oracle:oinstall /u01/client"
[root@orach1 ~]# podman exec oracc1 /bin/bash -c "chown -R oracle:oinstall /u01/oraInventory"
```

Set passwords for root and oracle on RAC and client hosts

You need to set the “root” and “oracle” passwords on all RAC and client hosts.

On Podman host orach1:

```
[root@orach1 ~]# podman exec oraph1 /bin/bash -c "echo "welcome" | passwd root --stdin"
[root@orach1 ~]# podman exec oraph1 /bin/bash -c "echo "oracle" | passwd oracle --stdin"
#
[root@orach1 ~]# podman exec oraph2 /bin/bash -c "echo "welcome" | passwd root --stdin"
[root@orach1 ~]# podman exec oraph2 /bin/bash -c "echo "oracle" | passwd oracle --stdin"
#
[root@orach1 ~]# podman exec oracc1 /bin/bash -c "echo "welcome" | passwd root --stdin"
[root@orach1 ~]# podman exec oracc1 /bin/bash -c "echo "oracle" | passwd oracle --stdin"
```

On Podman host orach2:

```
[root@orach2 ~]# podman exec oraph3 /bin/bash -c "echo "welcome" | passwd root --stdin"
[root@orach2 ~]# podman exec oraph3 /bin/bash -c "echo "oracle" | passwd oracle --stdin"
#
[root@orach2 ~]# podman exec oraph4 /bin/bash -c "echo "welcome" | passwd root --stdin"
[root@orach2 ~]# podman exec oraph4 /bin/bash -c "echo "oracle" | passwd oracle --stdin"
```

Set up passwordless SSH

Refer to the Oracle Grid Infrastructure Installation and Upgrade Guide for Linux for information on configuring Secure Shell (SSH) and user equivalency.

Configure remote display of primary RAC host oraph1

Note: When using macvlan, you will find that a Podman host will not be able to connect to its RAC hosts and the RAC hosts will not be able to connect their Podman hosts). This is a macvlan limitation. Refer to the USING DOCKER MACVLAN NETWORKS page that describes the issue and a work-around (that we don't need to do).

Start Virtual Network Computing (VNC) server on the Podman host orach2:

```
[root@orach2 ~]# vncserver
WARNING: vncserver has been replaced by a systemd unit and is now considered deprecated and removed
in upstream.
Please read /usr/share/doc/tigervnc/HOWTO.md for more information.
New 'orach2:1 (root)' desktop is orach2:1
Starting applications specified in /root/.vnc/xstartup
Log file is /root/.vnc/orach2:1.log
```

Change the `sshd` settings (in /etc/ssh/sshd_config) on the client oraph1:

```
X11Forwarding yes
X11DisplayOffset 10
```

```
X11UseLocalhost no
```

Restart `sshd` on the client oraph1:

```
[root@oraph1 ~]# systemctl daemon-reload
[root@oraph1 ~]# systemctl restart sshd
```

Using some VNC client app, connect to the VNC server running on Podman host, orach2.
For example, in this case, it is orach2:1.

Open a terminal and connect to the client, oraph1

Add oraph1 (129.40.1.11) to the access control list (ACL):

```
[root@orach2 ~]# xhost + 129.40.1.11
129.40.1.11 being added to access control list
```

Connect to oraph1 (129.40.1.11) as user, oracle:

```
[root@orach2 ~]# ssh -X oracle@129.40.1.11
oracle@129.40.1.11's password:
Last login: Wed Oct 26 10:49:30 2022 from 129.40.1.2
[oracle@oraph1 ~]$
```

Test using the `xcterm` command to verify.

OK, now GUI apps can be run on oraph1 with output going to VNC client session on orach2.

Configure remote display of client host oracc1

Note: When using macvlan, you will find that a Podman host will not be able to connect to its RAC hosts and the RAC hosts will not be able to connect their Podman hosts). This is a macvlan limitation. Refer to the USING DOCKER MACVLAN NETWORKS page that describes the issue and a work-around (that we don't need to do).

Start the VNC server on the Podman host, orach2:

```
[root@orach2 ~]# vncserver
WARNING: vncserver has been replaced by a systemd unit and is now considered deprecated and removed
in upstream.
Please read /usr/share/doc/tigervnc/HOWTO.md for more information.
New 'orach2:1 (root)' desktop is orach2:1
Starting applications specified in /root/.vnc/xstartup
Log file is /root/.vnc/orach2:1.log
```

Change the `sshd` settings (in /etc/ssh/sshd_config) on the client oracc1:

```
X11Forwarding yes
X11DisplayOffset 10
X11UseLocalhost no
```

Restart `sshd` on the client oracc1:

```
[root@oracc1 ~]# systemctl daemon-reload
[root@oracc1 ~]# systemctl restart sshd
```

Using some VNC client app, connect to the VNC server running on Podman host, orach2. In this example, it is orach2:1.

Open a terminal and connect to the client, oracc1

Add oracc1 (129.40.1.24) to ACL:
```
[root@orach2 ~]# xhost + 129.40.1.24
129.40.1.24 being added to access control list
```

Connect to oracc1 (129.40.1.24) as user, oracle:
```
[root@orach2 ~]# ssh -X oracle@129.40.1.24
oracle@129.40.1.24's password:
Last login: Wed Oct 26 10:49:30 2022 from 129.40.1.2
[oracle@oracc1 ~]$
```

Test using the `xterm` command to verify.

Now, the GUI apps can be run on oracc1 with output going to VNC client session on orach2.

Install Oracle Grid Infrastructure

Note: This procedure in this entire section is performed on the oraph1 primary RAC host only.

Prepare the grid home for installation

As the *oracle* user, in the VNC terminal, on oraph1 extract the Oracle Grid Infrastructure files.

Extract the 19.16 maintenance as user, oracle:
```
[oracle@oraph1 ~]$ cd /software/stage
[oracle@oraph1 stage]$ unzip -q p34130714_190000_Linux-zSer.zip
```

Extract the 19.16 OJVM maintenance as user, oracle:
```
[oracle@oraph1 stage]$ unzip -q p34086870_190000_Linux-zSer.zip
```

Copy the 19.3 GI zip file to grid home:
```
[oracle@oraph1 stage]$ cd /u01/grid
[oracle@oraph1 grid]$ cp /software/stage/LINUX.ZSERIES64_193000_grid_home.zip .
[oracle@oraph1 grid]$ unzip -q LINUX.ZSERIES64_193000_grid_home.zip
```

Get the new OPatch:
```
[oracle@oraph1 grid]$ mv OPatch OPatch_19.3
[oracle@oraph1 grid]$ cp /software/stage/p6880880_180000_Linux-zSer.zip .
[oracle@oraph1 grid]$ unzip -q p6880880_180000_Linux-zSer.zip
```

Remove the zip files in grid home:
```
[oracle@oraph1 grid]$ rm -fr *.zip
```

Set the CVU environment variable for the current OS level

Set the environment variable so that cluvfy does not complain about the unknown OS:
```
[oracle@oraph1 grid]$ export CV_ASSUME_DISTID=RHEL8.6
```

Run runcluvfy to test

```
[oracle@oraph1 grid]$ /u01/grid/runcluvfy.sh stage -pre crsinst -n oraph1,oraph2,oraph3,oraph4

Note that you're running CVU against 19.3.
```

```
Many Oracle issues found are addressed in later maintenance.
This will probably get the following warnings/failures.
You can ignore these for now.

Verifying Package: cvuqdisk-1.0.10-1 ...FAILED
```
Note: Install cvuqdisk on all nodes
```
Verifying resolv.conf Integrity ...FAILED
```
Note: Oracle bug..can ignore
```
Verifying DNS/NIS name service ...FAILED
```
Note: Oracle bug..can ignore
```
Verifying /boot mount ...FAILED
```
Note: Oracle bug..can ignore
```
Verifying User Equivalence ...FAILED
```
Note: Oracle bug..can ignore
```
Verifying RPM Package Manager database ...INFORMATION
Verifying /dev/shm mounted as temporary file system ...FAILED
```
Note: podman bug..ignore for now

Start the GI installation

Perform the tasks in this section as user, *oracle*.

This is (19.3 + 19.16 maint + 19.16 OJVM maint).

```
[oracle@oraph1 grid]$ /u01/grid/gridSetup.sh \
    -applyRU /software/stage/34130714 \
    -applyOneOffs /software/stage/34086870
Preparing the home to patch...
Applying the patch /software/stage/34130714...
Successfully applied the patch.
Applying the patch /software/stage/34086870...
Successfully applied the patch.
The log can be found at: /tmp/GridSetupActions2022-11-07_10-47-54AM/installerPatchActions_2022-11-
07_10-47-54AM.log
Launching Oracle Grid Infrastructure Setup Wizard...
The response file for this session can be found at:
 /u01/grid/install/response/grid_2022-11-07_10-47-54AM.rsp
You can find the log of this install session at:
 /tmp/GridSetupActions2022-11-07_10-47-54AM/gridSetupActions2022-11-07_10-47-54AM.log
Moved the install session logs to:
 /u01/oraInventory/logs/GridSetupActions2022-11-07_10-47-54AM

Save the logs to /home/oracle on oraph1:
[oracle@oraph1 grid]$ cd /home/oracle
[oracle@oraph1 ~]$ tar -cf Grid_Install.tar /u01/oraInventory/logs/GridSetupActions2022-11-07_10-
47-54AM
tar: Removing leading `/' from member names
[oracle@oraph1 ~]$ gzip Grid_Install.tar
[oracle@oraph1 ~]$ cksum Grid_Install.tar.gz
274224947 1577774 Grid_Install.tar.gz
```

Perform the grid installation

Select the following options in the GUI:

- Configure Grid Infrastructure for a New Cluster
  - Configure an Oracle Standalone Cluster

- Cluster name: oraph0104
- SCAN name: orascanpm14
- Public hostnames: oraph<1,2,3,4>.pbm.ihost.com
- Virtual hostnames: oraph<1,2,3,4>v.pbm.ihost.com
- eth0 is public
- eth1/2 are ASM & private
- Use Oracle Flex ASM for Storage
- Configure GIMR
- Separate ASM group for GIMR
- Create ASM disk group "DATA" for new ASM disk
- External redundancy
- Discovery Path: /dev/asm*
- Use disks asm-b816/17/18
- Create ASM disk group "MGMT" for GIMR
- External redundancy
- Discovery Path: /dev/asm*
- Use disks asm-b819/1a/1b
- Use password "oracle" for SYS and ASMSNMP
- Do not register with EM Cloud Control
- ASM Admin Group: asmadmin
- ASM DBA Group: asmdba
- ASM Operator Group: asmoper
- Oracle base: /u01/base
- Oracle Inventory: /u01/oraInventory
- Manually run Configuration Scripts
- In Prerequisite Checks window, select "Fix&Check Again" for any fixable issues
- Select "Install"
- Run the scripts presented on all nodes:
    - /u01/oraInventory/orainstRoot.sh
    - /u01/grid/root.sh
- The GUI app completes

Check installed levels

Verify that opatch shows we have installed required software at the 19.16 level

```
[oracle@oraph1 ~]$ export ORACLE_HOME=/u01/grid
[oracle@oraph1 ~]$ /u01/grid/OPatch/opatch lspatches
34086870;OJVM RELEASE UPDATE: 19.16.0.0.220719 (34086870)
34160635;OCW RELEASE UPDATE 19.16.0.0.0 (34160635)
34133642;Database Release Update : 19.16.0.0.220719 (34133642)
OPatch succeeded.
```

Install Oracle RDBMS

Note: The procedure in this entire section is performed on the oraph1 primary node only.

Prepare *oracle home* for installation

As the *oracle* user, in the VNC terminal, on oraph1 extract the Oracle RDBMS infrastructure files:

```
# Copy 19.3 RDBMS zip file to oracle home
[oracle@oraph1 ~]$ cd /u01/db
```

```
[oracle@oraph1 db]$ cp /software/stage/LINUX.ZSERIES64_193000_db_home.zip .
[oracle@oraph1 db]$ unzip -q LINUX.ZSERIES64_193000_db_home.zip
# get the new OPatch
[oracle@oraph1 db]$ mv OPatch OPatch_19.3
[oracle@oraph1 db]$ cp /software/stage/p6880880_180000_Linux-zSer.zip .
[oracle@oraph1 db]$ unzip -q p6880880_180000_Linux-zSer.zip
# remove the zip files in the oracle home
[oracle@oraph1 db]$ rm -fr *.zip
```

Set CVU env var for current OS level

Set the environment variable so that cluvfy does not complain about the unknown OS.

```
[oracle@oraph1 db]$ export CV_ASSUME_DISTID=RHEL8.6
```

Alternately, set this in the Oracle user.bash_profile.

Start the RDBMS installation

As user, "oracle":
This is (19.3 + 19.16 maint + 19.16 OJVM maint)

```
[oracle@oraph1 db]$ /u01/db/runInstaller \
   -applyRU /software/stage/34130714 \
   -applyOneOffs /software/stage/34086870
Preparing the home to patch...
Applying the patch /software/stage/34130714...
Successfully applied the patch.
Applying the patch /software/stage/34086870...
Successfully applied the patch.
The log can be found at: /u01/oraInventory/logs/InstallActions2022-11-07_01-13-
39PM/installerPatchActions_2022-11-07_01-13-39PM.log
Launching Oracle Database Setup Wizard...
The response file for this session can be found at:
 /u01/db/install/response/db_2022-11-07_01-13-39PM.rsp
You can find the log of this install session at:
 /u01/oraInventory/logs/InstallActions2022-11-07_01-13-39PM/installActions2022-11-07_01-13-39PM.log
```

Save the logs to /home/oracle on oraph1:

```
[oracle@oraph1 db]$ cd /home/oracle
[oracle@oraph1 ~]$ tar -cf DB_Install.tar /u01/oraInventory/logs/InstallActions2022-11-07_01-13-
39PM
tar: Removing leading `/' from member names
[oracle@oraph1 ~]$ gzip DB_Install.tar
[oracle@oraph1 ~]$ cksum DB_Install.tar.gz
585409476 1480099 DB_Install.tar.gz
```

Perform the RDBMS installation

Select the following options in the GUI:

- Set Up Software Only
- Oracle Real Application Clusters database installation
- select all nodes
- Enterprise Edition
- Oracle base: /u01/base (chmod 775 /u01/base)
- Select "dba" for all OS groups

- Install
- Run the scripts presented on all nodes:
    - /u01/db/root.sh
- The GUI app completes

Check installed levels

Using opatch, verify that you have installed the required software at the 19.16 level.

```
[oracle@oraph1 ~]$ export ORACLE_HOME=/u01/db
[oracle@oraph1 ~]$ /u01/db/OPatch/opatch lspatches
34160635;OCW RELEASE UPDATE 19.16.0.0.0 (34160635)
34086870;OJVM RELEASE UPDATE: 19.16.0.0.220719 (34086870)
34133642;Database Release Update : 19.16.0.0.220719 (34133642)
```

Install Oracle RDBMS client

Note: This procedure in this entire section is performed on the oracc1 primary node only.

Prepare ORACLE HOME for installation

As *oracle* user, in the VNC terminal, on oracc1 extract the Oracle RDBMS client files:

```
# Copy 19.3 RDBMS client zip file to tmp
[oracle@oracc1 ~]$ su
Password: <root_passwd>
[root@oracc1 ~]# mkdir /u01/tmp
[root@oracc1 ~]# chown oracle:oinstall /u01/tmp
[root@oracc1 ~]# exit
[oracle@oracc1 ~]$ cd /u01/tmp
[oracle@oracc1 tmp]$ cp /software/stage/LINUX.ZSERIES64_193000_client.zip .
[oracle@oracc1 tmp]$ unzip -q LINUX.ZSERIES64_193000_client.zip
[oracle@oracc1 tmp]$ rm -fr *.zip
```

Set the CVU environment variable for the current OS level

Set the environment variable so that cluvfy does not complain about the unknown OS:

```
[oracle@oracc1 tmp]$ export CV_ASSUME_DISTID=RHEL8.6
```

Start the client installation

This is (19.3 + 19.16 maint + 19.16 OJVM maint).

```
[oracle@oracc1 tmp]$ cd /u01/tmp/client
[oracle@oracc1 client]$ ./runInstaller
Starting Oracle Universal Installer...
Checking Temp space: must be greater than 174 MB.   Actual 644875 MB    Passed
Checking swap space: must be greater than 150 MB.   Actual 42259 MB    Passed
Checking monitor: must be configured to display at least 256 colors.    Actual 16777216    Passed
Preparing to launch Oracle Universal Installer from /tmp/OraInstall2022-11-08_11-10-53AM. Please
wait ...[oracle@oracc1 client]$ The response file for this session can be found at:
 /u01/client/install/response/client_2022-11-08_11-10-53AM.rsp
The log of this install session can be found at:
 /u01/oraInventory/logs/installActions2022-11-08_11-10-53AM.log
```

```
[oracle@oracc1 client]$
```

Perform the client installation

Select the following options in the GUI

- Select **Administrator**.
- Select **Oracle base: /u01/base**.
- Select **Software location: /u01/client**.
- Select **Inventory Directory: /u01/oraInventory**.
- Select **Install**.

Then, run the scripts presented on all nodes:
/u01/oraInventory/orainstRoot.sh

The GUI app completes.

Test the Podman RAC environment

Test various Podman display type commands

Check Podman objects

```
[root@orach1 ~]# podman images
REPOSITORY                                TAG         IMAGE ID      CREATED       SIZE
localhost/oracle/database-rac             8.6-19.16   3356599322b8  10 days ago   1.17 GB
registry.access.redhat.com/ubi8/ubi       8.6-943     e683387831b3  3 months ago  225 MB
```

Note: You can see that the base Red Hat 8.6 image is there, and the new Red Hat 8.6 image, additional packages, and config is also there.

```
[root@orach1 ~]# podman network ls
NETWORK ID    NAME        DRIVER
2d871d1285fe  pm_priv1    macvlan
7dd5d7263eb7  pm_priv2    macvlan
34da88cc1137  pm_pub1     macvlan
2f259bab93aa  podman      bridge
```

Notice that all macvlan networks are defined.

```
[root@orach1 ~]# podman container list
CONTAINER ID  IMAGE                                      COMMAND               CREATED       STATUS        PORTS   NAMES
57524c91f8e1  localhost/oracle/database-rac:8.6-19.16     /usr/sbin/init        10 days ago   Up 3 days ago         oraph1
b5e56f48cd65  localhost/oracle/database-rac:8.6-19.16     /usr/sbin/init        10 days ago   Up 3 days ago         oraph2
2a610557e1be  localhost/oracle/database-rac:8.6-19.16     /usr/sbin/init        19 hours ago  Up 19 hours ago       oracc1
```

Notice that the Podman host, orach1, has containers oraph1, oraph2, and oracc1.

```
[root@orach2 ~]# podman container list
CONTAINER ID  IMAGE                                      COMMAND         CREATED      STATUS          PORTS       NAMES
c64425190626  localhost/oracle/database-rac:8.6-19.16  /usr/sbin/init  10 days ago  Up 3 days ago               oraph3
d594e1513cdb  localhost/oracle/database-rac:8.6-19.16  /usr/sbin/init  10 days ago  Up 3 days ago               oraph4
```

Notice that the Podman host, orach2, has containers oraph3 and oraph4.

```
[root@orach1 ~]# podman stats --no-reset
ID            NAME       CPU %       MEM USAGE / LIMIT  MEM %       NET IO            BLOCK IO           PIDS        CPU TIME           AVG CPU %
2a610557e1be  oracc1     0.12%       4.668GB / 17.18GB  27.17%      10.53MB / 3.89MB  2.47GB / 252.4MB   11          1m23.302104652s    0.12%
57524c91f8e1  oraph1     5.96%       7.071GB / 68.72GB  10.29%      45.83GB / 52.79GB  94.71GB / 29.13GB  666         5h37m5.827840824s  5.96%
b5e56f48cd65  oraph2     4.25%       10.66GB / 68.72GB  15.52%      24.71GB / 22.06GB  28.26GB / 10.05GB  539         4h0m23.389366268s  4.25%

ID            NAME       CPU %       MEM USAGE / LIMIT  MEM %       NET IO            BLOCK IO           PIDS        CPU TIME           AVG CPU %
2a610557e1be  oracc1     0.00%       4.668GB / 17.18GB  27.17%      10.53MB / 3.89MB  2.47GB / 252.4MB   11          1m23.302249544s    0.12%
57524c91f8e1  oraph1     5.75%       7.071GB / 68.72GB  10.29%      45.84GB / 52.8GB   94.71GB / 29.13GB  666         5h37m6.115586874s  5.96%
b5e56f48cd65  oraph2     6.38%       10.66GB / 68.72GB  15.52%      24.71GB / 22.06GB  28.26GB / 10.05GB  539         4h0m23.708739779s  4.25%
```

```
...
```

Notice that the Podman stats looks functional.

Check RAC host oraph1 running on Podman host orach1

```
[root@orach1 ~]# podman exec -it oraph1 bash
```

From Podman host, orach1, connect to RAC host, oraph1:

```
[root@oraph1 ~]# hostname
oraph1
```

Notice that the hostname is correct.

```
[root@oraph1 ~]# df -h /dev/shm
Filesystem      Size  Used Avail Use% Mounted on
shm              12G  637M   12G   6% /dev/shm
```

/dev/shm is what we specified in the container definition.

```
[root@oraph1 ~]# su - oracle
Last login: Mon Dec 19 15:13:21 PST 2022 on pts/2
[oracle@oraph1 ~]$ id
uid=54321(oracle) gid=54321(oinstall)
groups=54321(oinstall),54322(dba),54327(asmdba),54328(asmoper),54329(asmadmin)
```

Notice that the *oracle* user is assigned to the correct groups

Check kernel parms:

```
[root@oraph1 ~]# sysctl -a | grep kernel.sem
kernel.sem = 250        32000   100     128
kernel.sem_next_id = -1
[root@oraph1 ~]# sysctl -a | grep kernel.shmall
kernel.shmall = 8388608
[root@oraph1 ~]# sysctl -a | grep kernel.shmmax
kernel.shmmax = 34359738368
[root@oraph1 ~]# sysctl -a | grep kernel.shmmni
kernel.shmmni = 4096
[root@oraph1 ~]# sysctl -a | grep net.ipv4.ip_local_port_range
net.ipv4.ip_local_port_range = 9000     65500
[root@oraph1 ~]# sysctl -a | grep net.ipv4.conf.eth1.rp_filter
net.ipv4.conf.eth1.rp_filter = 2
[root@oraph1 ~]# sysctl -a | grep net.ipv4.conf.eth2.rp_filter
net.ipv4.conf.eth2.rp_filter = 2
```

Notice that the kernel parms look fine.

```
[root@oraph1 ~]# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
2: eth1@if4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 8992 qdisc noqueue state UP group default
    link/ether 5e:7a:c4:93:a2:44 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 12.1.1.201/24 brd 12.1.1.255 scope global eth1
```

```
       valid_lft forever preferred_lft forever
    inet 169.254.5.143/20 brd 169.254.15.255 scope global eth1:1
       valid_lft forever preferred_lft forever
3: eth2@if3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 8992 qdisc noqueue state UP group default
    link/ether 26:ed:72:0b:cb:2b brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 13.1.1.201/24 brd 13.1.1.255 scope global eth2
       valid_lft forever preferred_lft forever
    inet 169.254.26.149/20 brd 169.254.31.255 scope global eth2:1
       valid_lft forever preferred_lft forever
4: eth0@if2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether ca:a1:33:40:fb:78 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 129.40.1.11/27 brd 129.40.1.31 scope global eth0
       valid_lft forever preferred_lft forever
    inet 129.40.1.5/27 brd 129.40.1.31 scope global secondary eth0:1
       valid_lft forever preferred_lft forever
    inet 129.40.1.20/27 brd 129.40.1.31 scope global secondary eth0:4
       valid_lft forever preferred_lft forever
```

Note:

- The NIC names look odd, but this is normal for a container image host.
- The HAIPs look good.

```
[root@oraph1 ~]# ip route
default via 129.40.1.30 dev eth0
12.1.1.0/24 dev eth1 proto kernel scope link src 12.1.1.201
13.1.1.0/24 dev eth2 proto kernel scope link src 13.1.1.201
129.40.1.0/27 dev eth0 proto kernel scope link src 129.40.1.11
169.254.0.0/20 dev eth1 proto kernel scope link src 169.254.1.46
169.254.16.0/20 dev eth2 proto kernel scope link src 169.254.19.7
```

Notice that the routing looks fine.

```
[root@oraph1 ~]# ls -al /dev/asm*
brw-rw---- 1 oracle asmadmin 94,  41 Dec 20 10:05 /dev/asm-b816
brw-rw---- 1 oracle asmadmin 94,  45 Dec 16 11:47 /dev/asm-b817
brw-rw---- 1 oracle asmadmin 94,  49 Dec 20 08:04 /dev/asm-b818
brw-rw---- 1 oracle asmadmin 94,  53 Dec 20 10:05 /dev/asm-b819
brw-rw---- 1 oracle asmadmin 94,  57 Dec 20 10:05 /dev/asm-b81a
brw-rw---- 1 oracle asmadmin 94,  61 Dec 20 10:05 /dev/asm-b81b
brw-rw---- 1 oracle asmadmin 94,  65 Dec 20 10:05 /dev/asm-b81c
brw-rw---- 1 oracle asmadmin 94,  69 Dec 20 10:05 /dev/asm-b81d
brw-rw---- 1 oracle asmadmin 94,  73 Dec 20 10:05 /dev/asm-b81e
brw-rw---- 1 oracle asmadmin 94,  77 Dec 16 11:46 /dev/asm-b81f
brw-rw---- 1 oracle asmadmin 94,  81 Dec 16 11:46 /dev/asm-b820
brw-rw---- 1 oracle asmadmin 94,  85 Dec 16 11:46 /dev/asm-b821
brw-rw---- 1 oracle asmadmin 94,  89 Dec 16 11:46 /dev/asm-b822
brw-rw---- 1 oracle asmadmin 94,  93 Dec 16 11:46 /dev/asm-b823
brw-rw---- 1 oracle asmadmin 94,  97 Dec 16 11:46 /dev/asm-b824
brw-rw---- 1 oracle asmadmin 94, 101 Dec 16 11:46 /dev/asm-b825
```

There are shared devices and the permissions are good.

Check Grid Infrastructure

This section explains how to verify the basic functionality of Oracle GI.

Test 1 - Verify Oracle Cluster Ready Services (CRS) startup at boot time

Reboot all nodes and verify that CRS came up successfully. For this:

1. Stop the CRS cluster:
   /u01/grid/bin/crsctl stop cluster -all
2. On each of the following nodes, perform the following tasks:
   /u01/grid/bin/crsctl stop crs
   /u01/grid/bin/tfactl stop
   a. Stop all containers.
   b. Reboot Podman hosts, orach1 and orach2.
   c. Start all containers.

Check CRS.

```
[oracle@oraph1 ~]$ /u01/grid/bin/crsctl check cluster -all
**************************************************************
oraph1:
CRS-4537: Cluster Ready Services is online
CRS-4529: Cluster Synchronization Services is online
CRS-4533: Event Manager is online
**************************************************************
oraph2:
CRS-4537: Cluster Ready Services is online
CRS-4529: Cluster Synchronization Services is online
CRS-4533: Event Manager is online
**************************************************************
oraph3:
CRS-4537: Cluster Ready Services is online
CRS-4529: Cluster Synchronization Services is online
CRS-4533: Event Manager is online
**************************************************************
oraph4:
CRS-4537: Cluster Ready Services is online
CRS-4529: Cluster Synchronization Services is online
CRS-4533: Event Manager is online
**************************************************************
```

Check VIPs and HAIPs:

```
[oracle@oraph1 ~]$ ip addr | egrep 'eth0|eth1|eth2';echo ''
2: eth1@if4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 8992 qdisc noqueue state UP group default
    inet 12.1.1.201/24 brd 12.1.1.255 scope global eth1
    inet 169.254.1.46/20 brd 169.254.15.255 scope global eth1:1
3: eth2@if3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 8992 qdisc noqueue state UP group default
    inet 13.1.1.201/24 brd 13.1.1.255 scope global eth2
    inet 169.254.19.7/20 brd 169.254.31.255 scope global eth2:1
4: eth0@if2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    inet 129.40.1.11/27 brd 129.40.1.31 scope global eth0
    inet 129.40.1.5/27 brd 129.40.1.31 scope global secondary eth0:1
    inet 129.40.1.22/27 brd 129.40.1.31 scope global secondary eth0:2
```

Notice that HAIPs are fine on eth1 and eth2, VIP and SCAN on eth0.

```
[oracle@oraph2 ~]$ ip addr | egrep 'eth0|eth1|eth2';echo ''
2: eth1@if4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 8992 qdisc noqueue state UP group default
    inet 12.1.1.202/24 brd 12.1.1.255 scope global eth1
    inet 169.254.8.1/20 brd 169.254.15.255 scope global eth1:1
3: eth2@if3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 8992 qdisc noqueue state UP group default
```

```
    inet 13.1.1.202/24 brd 13.1.1.255 scope global eth2
    inet 169.254.29.103/20 brd 169.254.31.255 scope global eth2:1
4: eth0@if2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    inet 129.40.1.12/27 brd 129.40.1.31 scope global eth0
    inet 129.40.1.6/27 brd 129.40.1.31 scope global secondary eth0:5
```

Notice that HAIPs and fine on eth1 and eth2, and VIP on eth0.

```
[oracle@oraph3 ~]$ ip addr | egrep 'eth0|eth1|eth2';echo ''2: eth1@if2:
<BROADCAST,MULTICAST,UP,LOWER_UP> mtu 8992 qdisc noqueue state UP group default
    inet 12.1.1.203/24 brd 12.1.1.255 scope global eth1
    inet 169.254.14.26/20 brd 169.254.15.255 scope global eth1:1
3: eth2@if3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 8992 qdisc noqueue state UP group default
    inet 13.1.1.203/24 brd 13.1.1.255 scope global eth2
    inet 169.254.22.195/20 brd 169.254.31.255 scope global eth2:1
4: eth0@if4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    inet 129.40.1.13/27 brd 129.40.1.31 scope global eth0
    inet 129.40.1.7/27 brd 129.40.1.31 scope global secondary eth0:1
    inet 129.40.1.20/27 brd 129.40.1.31 scope global secondary eth0:2
```

Notice that HAIPs are fine on eth1 and eth2, VIP and SCAN on eth0.

```
[oracle@oraph4 ~]$ ip addr | egrep 'eth0|eth1|eth2';echo ''
2: eth2@if3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 8992 qdisc noqueue state UP group default
    inet 13.1.1.204/24 brd 13.1.1.255 scope global eth2
    inet 169.254.19.143/20 brd 169.254.31.255 scope global eth2:1
3: eth0@if4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    inet 129.40.1.14/27 brd 129.40.1.31 scope global eth0
    inet 129.40.1.8/27 brd 129.40.1.31 scope global secondary eth0:1
    inet 129.40.1.21/27 brd 129.40.1.31 scope global secondary eth0:2
4: eth1@if2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 8992 qdisc noqueue state UP group default
    inet 12.1.1.204/24 brd 12.1.1.255 scope global eth1
    inet 169.254.10.16/20 brd 169.254.15.255 scope global eth1:1
```

Notice that HAIPs are on eth1 and eth2, and VIP and SCAN on eth0.

Overall, test 1 passes.

Test 2 - Verify manual CRS startup

1. Set CRS to not start at boot time on all nodes as root user:
   `/u01/grid/bin/crsctl disable crs`

2. Stop the CRS cluster:
   `/u01/grid/bin/crsctl stop cluster -all`

3. On each of the following nodes, perform the following tasks as root user:
   `/u01/grid/bin/crsctl stop crs`

4. Verify CRS is down on all RAC hosts
   ```
   [root@oraph1 ~]# /u01/grid/bin/crsctl check crs

   CRS-4639: Could not contact Oracle High Availability Services

   [root@oraph2 ~]# /u01/grid/bin/crsctl check crs

     CRS-4639: Could not contact Oracle High Availability Services
   ```

```
[root@oraph3 ~]# /u01/grid/bin/crsctl check crs

  CRS-4639: Could not contact Oracle High Availability Services

[root@oraph4 ~]# /u01/grid/bin/crsctl check crs

  CRS-4639: Could not contact Oracle High Availability Services
```

Notice that CRS is down on all the nodes.

5. Start CRS and verify that it gets started successfully.

```
[root@oraph1 ~]# /u01/grid/bin/crsctl start crs

  CRS-4123: Oracle High Availability Services has been started.
```

Note that CRS started on oraph1.

```
[root@oraph1 ~]# /u01/grid/bin/crsctl check crs
  CRS-4638: Oracle High Availability Services is online
  CRS-4537: Cluster Ready Services is online
  CRS-4529: Cluster Synchronization Services is online
  CRS-4533: Event Manager is online
```

Notice that CRS is up on oraph1.

```
[root@oraph2 ~]# /u01/grid/bin/crsctl start crs

  CRS-4123: Oracle High Availability Services has been started.
```

Note that CRS started on oraph2.

```
[root@oraph2 ~]# /u01/grid/bin/crsctl check crs
  CRS-4638: Oracle High Availability Services is online
  CRS-4537: Cluster Ready Services is online
  CRS-4529: Cluster Synchronization Services is online
  CRS-4533: Event Manager is online
```

Notice that CRS is up on oraph2.

```
[root@oraph3 ~]# /u01/grid/bin/crsctl start crs

  CRS-4123: Oracle High Availability Services has been started.
```

Note that CRS started on oraph3.

```
[root@oraph3 ~]# /u01/grid/bin/crsctl check crs
  CRS-4638: Oracle High Availability Services is online
  CRS-4537: Cluster Ready Services is online
  CRS-4529: Cluster Synchronization Services is online
  CRS-4533: Event Manager is online
```

Notice that CRS is up on oraph3.

```
[root@oraph4 ~]# /u01/grid/bin/crsctl start crs

  CRS-4123: Oracle High Availability Services has been started.
```

Note that CRS started on oraph4.

```
[root@oraph4 ~]# /u01/grid/bin/crsctl check crs
  CRS-4638: Oracle High Availability Services is online
  CRS-4537: Cluster Ready Services is online
  CRS-4529: Cluster Synchronization Services is online
```

```
CRS-4533: Event Manager is online
```
Notice that CRS is up on oraph4.

```
[oracle@oraph1 ~]$ /u01/grid/bin/crsctl check cluster -all
**************************************************************
oraph1:
CRS-4537: Cluster Ready Services is online
CRS-4529: Cluster Synchronization Services is online
CRS-4533: Event Manager is online
**************************************************************
oraph2:
CRS-4537: Cluster Ready Services is online
CRS-4529: Cluster Synchronization Services is online
CRS-4533: Event Manager is online
**************************************************************
oraph3:
CRS-4537: Cluster Ready Services is online
CRS-4529: Cluster Synchronization Services is online
CRS-4533: Event Manager is online
**************************************************************
oraph4:
CRS-4537: Cluster Ready Services is online
CRS-4529: Cluster Synchronization Services is online
CRS-4533: Event Manager is online
**************************************************************
```
Note: The cluster is back up

6. Check VIPs and HAIPs:

```
[oracle@oraph1 ~]$ ip addr | egrep 'eth0|eth1|eth2';echo ''

2: eth1@if4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 8992 qdisc noqueue state UP group default
    inet 12.1.1.201/24 brd 12.1.1.255 scope global eth1
    inet 169.254.1.46/20 brd 169.254.15.255 scope global eth1:1
3: eth2@if3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 8992 qdisc noqueue state UP group default
    inet 13.1.1.201/24 brd 13.1.1.255 scope global eth2
    inet 169.254.19.7/20 brd 169.254.31.255 scope global eth2:1
4: eth0@if2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    inet 129.40.1.11/27 brd 129.40.1.31 scope global eth0
    inet 129.40.1.5/27 brd 129.40.1.31 scope global secondary eth0:3
```
Notice that HAIPs are fine on eth1 and eth2, and VIP on eth0

```
[oracle@oraph2 ~]$ ip addr | egrep 'eth0|eth1|eth2';echo ''

2: eth1@if4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 8992 qdisc noqueue state UP group default
    inet 12.1.1.202/24 brd 12.1.1.255 scope global eth1
    inet 169.254.8.1/20 brd 169.254.15.255 scope global eth1:1
3: eth2@if3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 8992 qdisc noqueue state UP group default
    inet 13.1.1.202/24 brd 13.1.1.255 scope global eth2
    inet 169.254.29.103/20 brd 169.254.31.255 scope global eth2:1
4: eth0@if2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    inet 129.40.1.12/27 brd 129.40.1.31 scope global eth0
    inet 129.40.1.6/27 brd 129.40.1.31 scope global secondary eth0:1
    inet 129.40.1.22/27 brd 129.40.1.31 scope global secondary eth0:2
```
Notice that HAIPs are fine on eth1 and eth2, and VIP SCAN on eth0.

```
[oracle@oraph3 ~]$ ip addr | egrep 'eth0|eth1|eth2';echo ''
```

```
2: eth2@if3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 8992 qdisc noqueue state UP group default
    inet 13.1.1.203/24 brd 13.1.1.255 scope global eth2
    inet 169.254.22.195/20 brd 169.254.31.255 scope global eth2:1
3: eth0@if4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    inet 129.40.1.13/27 brd 129.40.1.31 scope global eth0
    inet 129.40.1.7/27 brd 129.40.1.31 scope global secondary eth0:1
    inet 129.40.1.20/27 brd 129.40.1.31 scope global secondary eth0:2
4: eth1@if2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 8992 qdisc noqueue state UP group default
    inet 12.1.1.203/24 brd 12.1.1.255 scope global eth1
    inet 169.254.14.26/20 brd 169.254.15.255 scope global eth1:1
```

Notice that HAIPs are fine on eth1 and eth2, and VIP and SCAN on eth0.

```
[oracle@oraph4 ~]$ ip addr | egrep 'eth0|eth1|eth2';echo ''

2: eth2@if3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 8992 qdisc noqueue state UP group default
    inet 13.1.1.204/24 brd 13.1.1.255 scope global eth2
    inet 169.254.19.143/20 brd 169.254.31.255 scope global eth2:1
3: eth0@if4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    inet 129.40.1.14/27 brd 129.40.1.31 scope global eth0
    inet 129.40.1.8/27 brd 129.40.1.31 scope global secondary eth0:1
    inet 129.40.1.21/27 brd 129.40.1.31 scope global secondary eth0:2
4: eth1@if2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 8992 qdisc noqueue state UP group default
    inet 12.1.1.204/24 brd 12.1.1.255 scope global eth1
    inet 169.254.10.16/20 brd 169.254.15.255 scope global eth1:1
```

Notice that HAIPs are fine on eth1 and eth2, and VIP and SCAN on eth0.

7. Check /u01/grid/bin/crsctl status res -t

```
[oracle@oraph1 ~]$ /u01/grid/bin/crsctl status res -t
--------------------------------------------------------------------------------
Name           Target  State        Server                   State details
--------------------------------------------------------------------------------
Local Resources
--------------------------------------------------------------------------------
ora.LISTENER.lsnr
               ONLINE  ONLINE       oraph1                   STABLE
               ONLINE  ONLINE       oraph2                   STABLE
               ONLINE  ONLINE       oraph3                   STABLE
               ONLINE  ONLINE       oraph4                   STABLE
ora.net1.network
               ONLINE  ONLINE       oraph1                   STABLE
               ONLINE  ONLINE       oraph2                   STABLE
               ONLINE  ONLINE       oraph3                   STABLE
               ONLINE  ONLINE       oraph4                   STABLE
ora.ons
               ONLINE  ONLINE       oraph1                   STABLE
               ONLINE  ONLINE       oraph2                   STABLE
               ONLINE  ONLINE       oraph3                   STABLE
               ONLINE  ONLINE       oraph4                   STABLE
--------------------------------------------------------------------------------
Cluster Resources
--------------------------------------------------------------------------------
ora.ASMNET1LSNR_ASM.lsnr(ora.asmgroup)
      1        ONLINE  ONLINE       oraph1                   STABLE
      2        ONLINE  ONLINE       oraph2                   STABLE
      3        ONLINE  ONLINE       oraph3                   STABLE
ora.ASMNET2LSNR_ASM.lsnr(ora.asmgroup)
      1        ONLINE  ONLINE       oraph1                   STABLE
```

```
      2        ONLINE  ONLINE       oraph2                   STABLE
      3        ONLINE  ONLINE       oraph3                   STABLE
ora.DATA.dg(ora.asmgroup)
      1        ONLINE  ONLINE       oraph1                   STABLE
      2        ONLINE  ONLINE       oraph2                   STABLE
      3        ONLINE  ONLINE       oraph3                   STABLE
ora.LISTENER_SCAN1.lsnr
      1        ONLINE  ONLINE       oraph2                   STABLE
ora.LISTENER_SCAN2.lsnr
      1        ONLINE  ONLINE       oraph3                   STABLE
ora.LISTENER_SCAN3.lsnr
      1        ONLINE  ONLINE       oraph4                   STABLE
ora.MGMT.dg(ora.asmgroup)
      1        ONLINE  ONLINE       oraph1                   STABLE
      2        ONLINE  ONLINE       oraph2                   STABLE
      3        ONLINE  ONLINE       oraph3                   STABLE
ora.MGMTLSNR
      1        ONLINE  ONLINE       oraph1                   169.254.1.46 12.1.1. 201
13.1.1.201,STABLE
ora.asm(ora.asmgroup)
      1        ONLINE  ONLINE       oraph1                   Started,STABLE
      2        ONLINE  ONLINE       oraph2                   Started,STABLE
      3        ONLINE  ONLINE       oraph3                   Started,STABLE
ora.asmnet1.asmnetwork(ora.asmgroup)
      1        ONLINE  ONLINE       oraph1                   STABLE
      2        ONLINE  ONLINE       oraph2                   STABLE
      3        ONLINE  ONLINE       oraph3                   STABLE
ora.asmnet2.asmnetwork(ora.asmgroup)
      1        ONLINE  ONLINE       oraph1                   STABLE
      2        ONLINE  ONLINE       oraph2                   STABLE
      3        ONLINE  ONLINE       oraph3                   STABLE
ora.cvu
      1        ONLINE  ONLINE       oraph1                   STABLE
ora.mgmtdb
      1        ONLINE  ONLINE       oraph1                   Open,STABLE
ora.oraph1.vip
      1        ONLINE  ONLINE       oraph1                   STABLE
ora.oraph2.vip
      1        ONLINE  ONLINE       oraph2                   STABLE
ora.oraph3.vip
      1        ONLINE  ONLINE       oraph3                   STABLE
ora.oraph4.vip
      1        ONLINE  ONLINE       oraph4                   STABLE
ora.scan1.vip
      1        ONLINE  ONLINE       oraph2                   STABLE
ora.scan2.vip
      1        ONLINE  ONLINE       oraph3                   STABLE
ora.scan3.vip
      1        ONLINE  ONLINE       oraph4                   STABLE
--------------------------------------------------------------------------------
```

Overall, notice that test 2 passes.

Test 3 - Test NIC failover

Fail a private NIC on Podman host and verify if the failover worked.

Check the initial HAIPs on oraph1 and oraph2:

```
[oracle@oraph1 ~]$ ip addr | egrep 'eth1@|eth2@|169.254'
2: eth1@if4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 8992 qdisc noqueue state UP group default
    inet 169.254.1.46/20 brd 169.254.15.255 scope global eth1:1
3: eth2@if3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 8992 qdisc noqueue state UP group default
    inet 169.254.19.7/20 brd 169.254.31.255 scope global eth2:1

[oracle@oraph2 ~]$ ip addr | egrep 'eth1@|eth2@|169.254'
2: eth1@if4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 8992 qdisc noqueue state UP group default
    inet 169.254.8.1/20 brd 169.254.15.255 scope global eth1:1
3: eth2@if3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 8992 qdisc noqueue state UP group default
    inet 169.254.29.103/20 brd 169.254.31.255 scope global eth2:1
```

Fail priv2 on Podman host, orach1:

```
[root@orach1 ~]# echo "0" > /sys/devices/qeth/0.0.1490/online

[root@orach1 ~]# ip addr show priv2
3: priv2: <BROADCAST,MULTICAST> mtu 8992 qdisc mq state DOWN group default qlen 1000
    link/ether ee:d2:aa:03:96:98 brd ff:ff:ff:ff:ff:ff
    inet 13.1.1.101/24 brd 13.1.1.255 scope global noprefixroute priv2
       valid_lft forever preferred_lft forever
```

Note that priv2 (1490) is down.

Check CRS, VIPs, and HAIPs:

```
[oracle@oraph1 ~]$ /u01/grid/bin/crsctl check crs
CRS-4638: Oracle High Availability Services is online
CRS-4537: Cluster Ready Services is online
CRS-4529: Cluster Synchronization Services is online
CRS-4533: Event Manager is online

[oracle@oraph1 ~]$ ip addr | egrep 'eth1@|eth2@|169.254'
2: eth1@if4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 8992 qdisc noqueue state UP group default
    inet 169.254.1.46/20 brd 169.254.15.255 scope global eth1:1
    inet 169.254.19.7/20 brd 169.254.31.255 scope global eth1:2
3: eth2@if3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 8992 qdisc noqueue state UP group default

[oracle@oraph2 ~]$ ip addr | egrep 'eth1@|eth2@|169.254'
2: eth1@if4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 8992 qdisc noqueue state UP group default
    inet 169.254.8.1/20 brd 169.254.15.255 scope global eth1:1
    inet 169.254.29.103/20 brd 169.254.31.255 scope global eth1:2
3: eth2@if3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 8992 qdisc noqueue state UP group default
```

Note that HAIPs failed over from eth2 to eth1.

```
[oracle@oraph1 ~]$ /u01/grid/bin/crsctl check cluster -all
**************************************************************
oraph1:
CRS-4537: Cluster Ready Services is online
CRS-4529: Cluster Synchronization Services is online
CRS-4533: Event Manager is online
**************************************************************
oraph2:
CRS-4537: Cluster Ready Services is online
CRS-4529: Cluster Synchronization Services is online
CRS-4533: Event Manager is online
**************************************************************
```

```
oraph3:
CRS-4537: Cluster Ready Services is online
CRS-4529: Cluster Synchronization Services is online
CRS-4533: Event Manager is online
************************************************************
oraph4:
CRS-4537: Cluster Ready Services is online
CRS-4529: Cluster Synchronization Services is online
CRS-4533: Event Manager is online
************************************************************
```

Note:
In ohasd_orarootagent_root_1.trc:
2022-10-27 23:02:36.802 : USRTHRD:1803532560: [INFO] {0:5:3} HAIP: Moving ip '169.254.19.7' from inf 'eth2' to inf 'eth1'. This indicates that the cluster is still good.

Recover priv2 on Podman host orach1:

```
[root@orach1 ~]# echo "1" > /sys/devices/qeth/0.0.1490/online

[root@orach1 ~]# ip addr show priv2
3: priv2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 8992 qdisc mq state UP group default qlen 1000
    link/ether ee:d2:aa:03:96:98 brd ff:ff:ff:ff:ff:ff
    inet 13.1.1.101/24 brd 13.1.1.255 scope global noprefixroute priv2
       valid_lft forever preferred_lft forever
```

Note: priv2 (1490) is the back up.

Check CRS, VIPs, and HAIPs:

```
[oracle@oraph1 ~]$ /u01/grid/bin/crsctl check crs
CRS-4638: Oracle High Availability Services is online
CRS-4537: Cluster Ready Services is online
CRS-4529: Cluster Synchronization Services is online
CRS-4533: Event Manager is online

[oracle@oraph1 ~]$ ip addr | egrep 'eth1@|eth2@|169.254'
2: eth1@if4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 8992 qdisc noqueue state UP group default
    inet 169.254.1.46/20 brd 169.254.15.255 scope global eth1:1
3: eth2@if3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 8992 qdisc noqueue state UP group default
    inet 169.254.19.7/20 brd 169.254.31.255 scope global eth2:1

[oracle@oraph2 ~]$ ip addr | egrep 'eth1@|eth2@|169.254'
2: eth1@if4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 8992 qdisc noqueue state UP group default
    inet 169.254.8.1/20 brd 169.254.15.255 scope global eth1:1
3: eth2@if3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 8992 qdisc noqueue state UP group default
    inet 169.254.29.103/20 brd 169.254.31.255 scope global eth2:1
==> HAIPs recovered back to eth2 .. GOOD

[oracle@oraph1 ~]$ /u01/grid/bin/crsctl check cluster -all
************************************************************
oraph1:
CRS-4537: Cluster Ready Services is online
CRS-4529: Cluster Synchronization Services is online
CRS-4533: Event Manager is online
************************************************************
oraph2:
CRS-4537: Cluster Ready Services is online
CRS-4529: Cluster Synchronization Services is online
CRS-4533: Event Manager is online
```

```
**************************************************************
oraph3:
CRS-4537: Cluster Ready Services is online
CRS-4529: Cluster Synchronization Services is online
CRS-4533: Event Manager is online
**************************************************************
oraph4:
CRS-4537: Cluster Ready Services is online
CRS-4529: Cluster Synchronization Services is online
CRS-4533: Event Manager is online
**************************************************************
```
These results indicate that the cluster is still good.

Overall, test 3 passes.

Test 4 - Test GI-related commands

Test some GI-related commands:

```
[root@oraph1 ~]# export PATH=$PATH:/u01/grid/bin
[root@oraph1 ~]# export ORACLE_HOME=/u01/grid

[root@oraph1 ~]# asmcmd lsdg
State    Type    Rebal  Sector  Logical_Sector  Block       AU  Total_MB  Free_MB  Req_mir_free_MB
Usable_file_MB  Offline_disks  Voting_files  Name
MOUNTED  EXTERN  N         4096            4096   4096  4194304    126768   126376                0
126376              0             Y  DATA/
MOUNTED  EXTERN  N         4096            4096   4096  4194304    126768   102648                0
102648              0             N  MGMT/

[root@oraph1 ~]# lsnrctl status
LSNRCTL for Linux: Version 19.0.0.0.0 - Production on 27-OCT-2022 16:19:24
Copyright (c) 1991, 2022, Oracle.  All rights reserved.
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=LISTENER)))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 19.0.0.0.0 - Production
Start Date                27-OCT-2022 22:57:29
Uptime                    0 days 0 hr. 21 min. 54 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File   /u01/grid/network/admin/listener.ora
Listener Log File         /u01/base/diag/tnslsnr/oraph1/listener/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=LISTENER)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=129.40.1.11)(PORT=1521)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=129.40.1.5)(PORT=1521)))
Services Summary...
Service "+ASM" has 1 instance(s).
  Instance "+ASM1", status READY, has 1 handler(s) for this service...
Service "+ASM_DATA" has 1 instance(s).
  Instance "+ASM1", status READY, has 1 handler(s) for this service...
Service "+ASM_MGMT" has 1 instance(s).
  Instance "+ASM1", status READY, has 1 handler(s) for this service...
The command completed successfully
```

```
[root@oraph1 ~]# ocrcheck -details
Status of Oracle Cluster Registry is as follows :
         Version                  :          4
         Total space (kbytes)     :     901284
         Used space (kbytes)      :      84520
         Available space (kbytes) :     816764
         ID                       : 2002813216
         Device/File Name         : +DATA/oraph0104/OCRFILE/registry.255.1119137911
                                    Device/File integrity check succeeded
                                    Device/File not configured
                                    Device/File not configured
                                    Device/File not configured
                                    Device/File not configured
         Cluster registry integrity check succeeded
         Logical corruption check succeeded

[root@oraph1 ~]# oraversion -compositeVersion
19.16.0.0.0

[root@oraph1 ~]# su - oracle
Last login: Thu Oct 27 16:09:50 PDT 2022
[oracle@oraph1 ~]$ export ORACLE_SID=+ASM1;export ORACLE_HOME=/u01/grid;/u01/grid/bin/sqlplus / as
sysasm
SQL*Plus: Release 19.0.0.0.0 - Production on Thu Oct 27 16:30:49 2022
Version 19.16.0.0.0
Copyright (c) 1982, 2022, Oracle.  All rights reserved.
Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.16.0.0.0
SQL> exit
Disconnected from Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.16.0.0.0

[root@oraph1 ~]# srvctl status ons
ONS ons is enabled.
ONS ons is running.

[root@oraph1 ~]# srvctl status asm
ASM is running on oraph2,oraph3,oraph1

[root@oraph1 ~]# srvctl status mgmtdb
Database is enabled
Instance -MGMTDB is running on node oraph1

[root@oraph1 ~]# srvctl config mgmtdb
Database unique name: _mgmtdb
Database name:
Oracle home: <CRS home>
Oracle user: grid
Spfile: +MGMT/_MGMTDB/PARAMETERFILE/spfile.275.1119201131
Password file: +MGMT/_MGMTDB/PASSWORD/pwd_mgmtdb.257.1119200311
Domain:
Start options: open
Stop options: immediate
Database role: PRIMARY
Management policy: AUTOMATIC
Type: Management
PDB name: GIMR_DSCREP_10
PDB service: GIMR_DSCREP_10
```

```
Cluster name: oraph0104
Database instance: -MGMTDB

[root@oraph1 ~]# srvctl status nodeapps
VIP 129.40.1.5 is enabled
VIP 129.40.1.5 is running on node: oraph1
VIP 129.40.1.6 is enabled
VIP 129.40.1.6 is running on node: oraph2
VIP 129.40.1.7 is enabled
VIP 129.40.1.7 is running on node: oraph3
VIP 129.40.1.8 is enabled
VIP 129.40.1.8 is running on node: oraph4
Network is enabled
Network is running on node: oraph1
Network is running on node: oraph2
Network is running on node: oraph3
Network is running on node: oraph4
ONS is enabled
ONS daemon is running on node: oraph1
ONS daemon is running on node: oraph2
ONS daemon is running on node: oraph3
ONS daemon is running on node: oraph4

[root@oraph1 ~]# oclumon manage -get MASTER
Master = oraph1

[root@oraph1 ~]# /u01/grid/jdk/bin/java -version
java version "1.8.0_311"
Java(TM) SE Runtime Environment (build 8.0.7.0 - pxz6480sr7-20211025_01(SR7))
IBM J9 VM (build 2.9, JRE 1.8.0 Linux s390x-64-Bit Compressed References 20211022_15212 (JIT
enabled, AOT enabled)
OpenJ9   - 6abb372
OMR      - b898db9
IBM      - 2f2c48b)
JCL - 20210930_01 based on Oracle jdk8u311-b11

[root@oraph1 ~]# tfactl print status
.-------------------------------------------------------------------------------------------.
| Host   | Status of TFA | PID   | Port | Version    | Build ID             | Inventory Status |
+--------+---------------+-------+------+------------+----------------------+------------------+
| oraph1 | RUNNING       |   425 | 5000 | 22.1.3.0.0 | 22130020220621134312 | COMPLETE         |
| oraph2 | RUNNING       | 38330 | 5000 | 22.1.3.0.0 | 22130020220621134312 | COMPLETE         |
| oraph3 | RUNNING       | 37111 | 5000 | 22.1.3.0.0 | 22130020220621134312 | COMPLETE         |
| oraph4 | RUNNING       | 36287 | 5000 | 22.1.3.0.0 | 22130020220621134312 | COMPLETE         |
'--------+---------------+-------+------+------------+----------------------+------------------'
```

Overall, test 4 passes.

Test 5 - Create an ASM disk group

Use the asmca command to create a new disk group, DATA1.

- Disks: asm-b81c asm-b81d asm-b81e asm-b81f
- External redundancy

This appeared to work. To check this, run the following command:

```
[root@oraph1 ~]# /u01/grid/bin/asmcmd lsdg
State    Type    Rebal  Sector  Logical_Sector  Block       AU  Total_MB  Free_MB  Req_mir_free_MB  Usable_file_MB  Offline_disks  Voting_files  Name
MOUNTED  EXTERN  N         4096            4096   4096  4194304    126768   126376                0          126376              0             Y  DATA/
MOUNTED  EXTERN  N         4096            4096   4096  4194304    169024   168804                0          168804              0             N  DATA1/
MOUNTED  EXTERN  N         4096            4096   4096  4194304    126768   102648                0          102648              0             N  MGMT/
```

Overall, test 5 passes.

Check RDBMS

This section explains how to verify the basic functionality of Oracle RDBMS.

Test 1 - Test creation of a DB

Using the dbca GUI, create a DB on the new ASM data group, DATA1, and perform a typical installation with the following settings:

- CDB/PDB: orcl/orcl_pdb
- DB files: +DATA1
- FRA: +DATA1
- ADMIN PW: oracle

Errors:

```
[DBT-11209] Current available memory is less than the required available
memory (516,060MB) for creating the database.
CauseÂ -Â Following nodes do not have required available memory :
Node:oraph4 Available memory:20.76GB (2.1768484E7KB)
Node:oraph2 Available memory:21.6677GB (2.2720248E7KB)
Node:oraph3 Available memory:27.6493GB (2.8992436E7KB)
Node:oraph1 Available memory:34.7831GB (3.6472736E7KB)

[DBT-11207] Specified SGA size is greater than the shmmax on the
system. The database creation might fail with "ORA-27125 - Unable to
create shared memory segment error".
ActionÂ -Â Specify SGA size lesser than or equal to the shmmax on the system.
```

Note: This is a bug. Oracle CVU is not calculating the available memory correctly. So, use **Advanced Configuration** to make SGA less than 20 GB for now. Notice that CDB/PDB is then created successfully.

```
[oracle@oraph1 ~]$ export ORACLE_HOME=/u01/db
[oracle@oraph1 ~]$ export PATH=/u01/db/bin:$PATH

[oracle@oraph1 ~]$ srvctl status db -db orcl
Instance orcl1 is running on node oraph1
Instance orcl2 is running on node oraph2
Instance orcl3 is running on node oraph3
Instance orcl4 is running on node oraph4

[oracle@oraph1 ~]$ srvctl config db -db orcl
Database unique name: orcl
Database name: orcl
Oracle home: /u01/db
Oracle user: oracle
Spfile: +DATA1/ORCL/PARAMETERFILE/spfile.287.1119322003
Password file: +DATA1/ORCL/PASSWORD/pwdorcl.256.1119320571
Domain: pbm.ihost.com
Start options: open
Stop options: immediate
Database role: PRIMARY
Management policy: AUTOMATIC
Server pools:
Disk Groups: DATA1
Mount point paths:
Services:
Type: RAC
```

```
Start concurrency:
Stop concurrency:
OSDBA group: dba
OSOPER group:
Database instances: orcl1,orcl2,orcl3,orcl4
Configured nodes: oraph1,oraph2,oraph3,oraph4
CSS critical: no
CPU count: 0
Memory target: 0
Maximum memory: 0
Default network number for database services:
Database is administrator managed

[oracle@oraph1 ~]$ export ORACLE_SID=orcl1

[oracle@oraph1 ~]$ sqlplus / as sysdba
SQL*Plus: Release 19.0.0.0.0 - Production on Fri Oct 28 19:52:11 2022
Version 19.16.0.0.0
Copyright (c) 1982, 2022, Oracle.  All rights reserved.
Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.16.0.0.0
SQL> show pdbs
    CON_ID CON_NAME                       OPEN MODE  RESTRICTED
---------- ------------------------------ ---------- ----------
         2 PDB$SEED                       READ ONLY  NO
         3 ORCL_PDB                       READ WRITE NO

SQL> alter session set container=ORCL_PDB;
Session altered.

SQL> show con_name
CON_NAME
------------------------------
ORCL_PDB

SQL> exit
Disconnected from Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.16.0.0.0
[oracle@oraph1 ~]$
```

This looks good, and overall, test 1 passes.

Test 2 - Test creation of a tablespace/table

Create a tablespace and a table on the ORCL DB:

```
[oracle@oraph1 ~]$ sqlplus / as sysdba
SQL*Plus: Release 19.0.0.0.0 - Production on Fri Oct 28 19:54:53 2022
Version 19.16.0.0.0
Copyright (c) 1982, 2022, Oracle.  All rights reserved.
Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.16.0.0.0

SQL> alter session set container=ORCL_PDB;
Session altered.
```

```
SQL> create tablespace mike_ts datafile '+DATA1';
Tablespace created.

SQL> create table mike_tbl (x NUMBER(5) PRIMARY KEY, y VARCHAR2(15) NOT NULL) tablespace mike_ts;
Table created.

Add records to the table:
SQL> INSERT INTO mike_tbl (x,y) VALUES (12345, 'Mike Morgan');
1 row created.

SQL> select * from mike_tbl;
         X Y
---------- ---------------
     12345 Mike Morgan

SQL> exit
Disconnected from Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.16.0.0.0

[oracle@oraph1 ~]$
```

This looks good, and test 2 passes.

Check RDBMS client

This section explains how to verify the basic functionality of an Oracle RDBMS client.

Test 1 - Check easy connection

Run `SQL*Plus` connect from client, oracc1, to PDB orcl_pdb:

```
[oracle@oracc1 ~]$ hostname
oracc1

[oracle@oracc1 ~]$ /u01/client/bin/sqlplus
system/oracle@//orascanpm14.pbm.ihost.com/orcl_pdb.pbm.ihost.com
SQL*Plus: Release 19.0.0.0.0 - Production on Mon Dec 19 15:16:28 2022
Version 19.3.0.0.0
Copyright (c) 1982, 2019, Oracle.  All rights reserved.
Last Successful login time: Tue Dec 13 2022 10:23:35 -08:00
Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.16.0.0.0

SQL> show con_name
CON_NAME
------------------------------
ORCL_PDB
SQL>
```

Overall, this looks good, and the test passes.

Set up Transparent Network Substrate (TNS) entries

On the server nodes (oraph1, oraph2, oraph3, and oraph4): /u01/db/network/admin/tnsnames.ora:

```
ORCL =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = orascanpm14)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = orcl.pbm.ihost.com)
    )
  )

ORCL_PDB =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = orascanpm14)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = orcl_pdb.pbm.ihost.com)
    )
  )
```

On client oracc1: /u01/client/network/admin/tnsnames.ora:

```
ORCL =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = orascanpm14)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = orcl.pbm.ihost.com)
    )
  )

ORCL_PDB =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = orascanpm14)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = orcl_pdb.pbm.ihost.com)
    )
  )
```

Test 2 - Test client connectivity with TNS

Test from client as user, *oracle*:

```
[oracle@oracc1 ~]$ export ORACLE_HOME=/u01/client
```

Try tnsping of the CDB running on oraph1-4:

```
[oracle@oracc1 ~]$ /u01/client/bin/tnsping orcl
TNS Ping Utility for Linux: Version 19.0.0.0.0 - Production on 08-NOV-2022 11:22:20
Copyright (c) 1997, 2019, Oracle.  All rights reserved.
Used parameter files:
/u01/client/network/admin/sqlnet.ora
Used TNSNAMES adapter to resolve the alias
Attempting to contact (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST = orascanpm14)(PORT = 1521))
(CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = orcl.pbm.ihost.com)))
OK (0 msec)
```

Try tnsping of the PDB running on oraph1-4:

```
[oracle@oracc1 ~]$ /u01/client/bin/tnsping orcl_pdb
TNS Ping Utility for Linux: Version 19.0.0.0.0 - Production on 08-NOV-2022 11:22:23
Copyright (c) 1997, 2019, Oracle.  All rights reserved.
Used parameter files:
/u01/client/network/admin/sqlnet.ora
Used TNSNAMES adapter to resolve the alias
Attempting to contact (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST = orascanpm14)(PORT = 1521))
(CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = orcl_pdb.pbm.ihost.com)))
OK (0 msec)
```

Run sqlplus connect from client to orcl_pdb and do some work:

```
[oracle@oracc1 ~]$ /u01/client/bin/sqlplus system/oracle@orcl_pdb
SQL*Plus: Release 19.0.0.0.0 - Production on Tue Nov 8 11:25:58 2022
Version 19.3.0.0.0
Copyright (c) 1982, 2019, Oracle.  All rights reserved.
Last Successful login time: Tue Nov 08 2022 11:24:52 -08:00
Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.16.0.0.0

SQL> show con_name
CON_NAME
------------------------------
ORCL_PDB
```

Test 3 - Create tablespace/table

```
SQL> create tablespace test_ts datafile '+DATA1';
Tablespace created.

SQL> create table test_tbl (x NUMBER(5) PRIMARY KEY, y VARCHAR2(15) NOT NULL) tablespace test_ts;
Table created.

SQL> INSERT INTO test_tbl (x,y) VALUES (12345, 'This is a test');
1 row created.

SQL> select * from test_tbl;
         X Y
---------- ---------------
     12345 This is a test

SQL> exit
Disconnected from Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.16.0.0.0
[oracle@oracc1 ~]$
```

Overall, this looks good and the client tests pass.

Summary

This paper is intended for assisting in implementing Oracle RAC with Podman. This paper provides documentation in the form of examples explaining how to implement Podman with Oracle RAC on IBM System Z. In almost all cases, there are reinforcing references included in each section. The “References” section in this paper contains links to general documentation and additional supporting documents.

About the authors

Michael Morgan is a Senior Oracle Consultant currently working on certifications of the Oracle database running Linux on IBM LinuxONE and IBM Z servers in the Oracle Redwood Shores office. Before joining IBM as a consultant in 2015, he worked at Oracle Corporation in various system administration, management, and development roles.

Srujan D. Jagarlamudi is a Senior Oracle Consultant currently working with Linux running on IBM LinuxONE and IBM Z server certifications in the Oracle Redwood Shores office. Before joining IBM as a consultant in 2015, he worked in several IT companies as a Senior Oracle DBA and Applications DBA and played a major role in upgrading and migration projects to Oracle databases and applications.

References

Real Application Clusters (19c) Installation Guide for Podman Oracle Linux x86-64
https://docs.oracle.com/en/database/oracle/oracle-database/19/racpd/index.html

How to build Oracle RAC container image and access in your environment
https://github.com/oracle/docker-images/blob/main/OracleDatabase/RAC/README.md

Oracle container registry
https://container-registry.oracle.com/ords/f?p=113:10::::::

Oracle Grid Infrastructure Installation and Upgrade Guide for Linux
https://docs.oracle.com/en/database/oracle/oracle-database/19/cwlin/index.html

Oracle Database Installation Guide for Linux
https://docs.oracle.com/en/database/oracle/oracle-database/19/ladbi/index.html

Oracle GI/RDBMS downloads
https://www.oracle.com/database/technologies/oracle-database-software-downloads.html

Podman reported issues:

Unable to run a container with --cpu-rt-runtime when in cgroupV2
https://github.com/containers/podman/issues/10278

Podman is not supported the ipvlan driver
https://github.com/containers/podman/issues/11881

No option to specify static IP during Podman network connect command
https://github.com/containers/podman/issues/10277

Oracle RAC on PODMAN on IBM System z - Released Versions and Known Issues (Doc ID 2924682.1) (A MOS userid is needed to access this file)
https://support.oracle.com/epmos/faces/DocumentDisplay?id=2924682.1

IBM Systems
3039 Cornwallis Road
RTP, NC 27709

Produced in the United States of America

Please recycle