Ansible Oracle 19c RAC installation for AIX/Power Systems

README – V1.1.0

Feb 10th, 2023

The Ansible® Oracle® RAC ASM collection installs Oracle RAC 19c on AIX® operating System running IBM® Power® Systems servers. It has been tested on 1, 2, 3, 4, 5, 7 and 8 nodes clusters. The version v1.1.0 collections is tested on 2-node RAC with AIX 7.2TL5 and AIX 7.3TL0SP1

Setting up an Oracle Application Clusters (RAC) on AIX involves setting up an AIX environment on the hosts that meet the RAC's specific requirements from kernel tunables, network attributes, shared disk attributes, passwordless to user equivalent ssh connections etc. The manual process to accomplish these tasks is tedious and error prone. During the Grid and Database install, the GUI frequently prompts for entering input that ties up the user for a long time. The whole installation can take two days for seasoned users.

With the help of Ansible Oracle RAC ASM collection, it takes typically 5 hours to complete a 4-node RAC installation, a tremendous time saving. It's completely hands-free and can consistently recreate Oracle RACs for other projects. The value of this collection helps your organization to improve significant productivity.

New in Version V1.1.0

- Added support for AIX 7.3
- Added support for Ansible Automation Platform v2 (AAP2)
- Tested on PowerVS LPARs

Assumptions of using this collection

- The user is familiar with Ansible and should have at least the basic knowledge on YAML for the purpose of setting up the variables to run the playbook. Refer to Red Hat Ansible Automation Platform Docs at https://docs.ansible.com/automation.html
- The user is familiar with AIX administration and Oracle RAC configuration environment requirements. Refer to Oracle Grid Infrastructure Installation and Upgrade Guide 19c for IBM AIX on Power Systems (64-bit) at https://docs.oracle.com/en/database/oracle/oracle-database/19/books.html
- The user is familiar with installing packages on the Operating Systems supported by Ansible.

  https://access.redhat.com/articles/3168091

Supported environment

- AIX 7.2 TL4 or later, AIX 7.3 TL0 SP1 or later. Tested on AIX 7.2 TL4 SP1, AIX 7.2 TL5 SP2 and AIX 7.3 TL0 SP1.
- Oracle 19c RAC Standalone Cluster configuration type only. Base release, 19.3 alone is not supported. Tested releases were 19.8, 19.11, 19.12, 19.14 and 19.17(using -applyRU option).
- Oracle database is installed using Software install only option on ACFS shared filesystem.
- Ansible 2.9 or above. Tested versions were Ansible 2.9.11 and 2.12.1.

RAC nodes configuration requirements

- Each RAC node should have AIX 7.2 TL4 SP1 or above freshly installed.
- Three types of networks for RAC should be appropriately configured and tested for connectivity. Each RAC node must have:
    - Internet access network for downloading software packages, which connects to AIX toolbox for RPM install.
    - Oracle public networks (if this network can access the Internet, the Internet access network does not need to be configured).
    - Oracle private networks. Two networks are dedicated to Oracle for Interconnect and ASM traffic. Separate ASM networks and Interconnect networks is not supported.
- All network interfaces must be consistent across the nodes. E.g. en0 on all nodes are connected to the same physical network, likewise for en1 and en2 etc. The playbook will perform consistency check and connectivity check on Oracle public networks and Oracle Interconnect networks.
- ASM shared disks are free of physical volume IDs (PVIDs), do not belong to any AIX volume groups nor has ASM disk group header. Refer to Appendix **Disk headers and PVIDs** for more information.
- ASM shared disks are consistent across all RAC nodes, for instance, hdisk10 on each node is indeed the same LUN off the storage. They playbook will check for shared disk consistency.
- Oracle Grid HOME is on a local filesystem created out of an AIX volume group, make sure the disks for the volume group do not have previous volume group associated with them.
- AIX filesets, Opatch zip files, Grid Infrastructure/Database install zip files, IBM XlC compiler fileset are available to the RAC nodes through NFS mount points.

DNS server(s) requirements

The RAC environment must have DNS server(s) for Domain Name resolution to resolve Oracle RAC SCAN name, Oracle Virtual IP addresses and for downloading software from the Internet. There are two choices:

- A DNS server that resolves Domain Name addresses for software download from the Internet, Oracle SCAN address, and Oracle Virtual IP addresses.
- Two DNS servers – one for resolving Domain Name addresses for software download from the Internet only, another for Oracle SCAN name and Oracle Virtual IP addresses. The two servers can be on different networks.

Ansible controller configuration requirements

- The Ansible controller has network connectivity to all RAC nodes. It has been tested with the controller and the RAC nodes connected over WAN/VPN as well as within the lab.
- Create a regular user such as 'ansible' for installing Ansible and running the playbook.
- Install python3 and python3-netaddr packages.
- The controller has wget, perl and expect RPMs installed.
- The controller has Ansible 2.9 or above installed. For details visit https://docs.ansible.com/ansible/latest/installation_guide/index.html
- Download the Ansible Oracle RAC collection and create a "top directory" for the collection.
- Install power_aix collection as the user who runs Ansible playbook.
  - $ ansible-galaxy collection install ibm.power_aix
  - For details, visit https://ibm.github.io/ansible-power-aix/installation.html
  - The collection is installed in ~/.ansible/collecrtions/ansible_collections/ibm/power_aix.
- The controller has access to the Internet for downloading software.
- Ansible Oracle RAC collection has three files that need to be modified for your environment:
  - ansible.cfg
  - inventory.yml – specifies the IP addresses of the RAC hosts.
  - vars.yml – specifies values for variables to configure the RAC hosts, AIX, Grid Infrastructure and RAC database.

Features

- The playbook has been developed with idempotency in mind. Idempotent means a task operation is only performed once, regardless the task is invoked many times thereafter. However, if a configuration variable that affects the outcome of the operation has changed after the operation has already been performed, it will perform the operation in most cases but not all. For instance, the playbook allows disks to be renamed to better indicate their intended purposes, but if the tasks prior to the renaming disks task need to be rerun maybe due to the disks have been renamed incorrectly. In this scenario, tasks that reference renamed disks will fail, and manual intervention is needed. Accommodating changes like this is a best effort attempt, therefore it's crucial to have the correct values for the variables the first time.

- As mentioned above, AIX hdiskX can optionally be renamed to <prefix>X or <prefix>Y, where Y is a different number from X and <prefix> is a user-specified string. This feature makes it easier to correlate the purpose of the disks in a large disk configuration.
- The recommended AIX tunable values and various settings are applied according to applicable Oracle documentation for running a RAC in the areas of networking, kernel, and disk attributes etc. They are built into the Ansible tasks. Refer to Oracle Grid Infrastructure Installation and Upgrade Guide 19c for IBM AIX on Power System (64-bit) at https://docs.oracle.com/en/database/oracle/oracle-database/19/books.html
- The dsm fileset for distributed shell (dsh) will be installed to ease the cluster administrative tasks. The collection also depends on it.
- IBM XL C 13 for AIX compiler installation is optional. It is installed on the named RAC host(s) specified in vars.yml. It requires the filesets to be accessible through NFS mount point.
- VNC server RPM will be installed on the RAC nodes when it is specified(optional). Users other than 'root' in the vars.yml will have VNC files created in ~/.vnc/passwd.
- Ansible uses ssh passwordless connections to drive the setup tasks on the RAC nodes. The bootstrap role (see below) creates the ssh passwordless configuration without the need to manually perform it beforehand. Ansible relies on the host's IP addresses and the root login password specified on the var.yml to create such configuration.
- AIX system files /etc/environment, /etc/hosts, /etc/security/limits, /etc/pam.conf and /etc/syslog.conf etc. are saved before they are modified. They are saved in <Ansible work directory>/saved directory, where <Ansible work directory> is specified in global section of vars.yml.

Playbook directory layout

The playbook is divided into the four roles:

- bootstrap – sets up the basic environment to enable full functionality of Ansible, set nameserver, binding and passwordless connections to the RAC nodes.
- preconfig – sets up basic environment such as time of day, configure for accessing Internet and consistent AIX version, release, TL, and SP. NFS mounts AIX filesets and installs the filesets.
- config – sets up AIX to meet the requirements for installing a RAC.
- Install – creates ASM disk groups, ACFS, prepares for installing Grid and database and finally install them.

Playbook top directory

```
        ├── ansible.cfg
        ├── inventory.yml
        ├── vars.yml
        ├── play.yml
        └── roles
            ├── bootstrap
            ├── preconfig
            ├── config
            └── install
```

Ansible variable file vars.yml

The vars.yml specifies how/what the Oracle RAC is to be configured.

The file is broadly divided by sections pertaining to the roles as listed above.

Comments are included to guide the users what values are set for the variables. Since this file is in YAML format, the dictionary key components leading to the leaf elements should make it apparent as to where the variable is intended and suggestive of appropriate value. Care should be made to assign the data value to the leaf elements, they can list structures. As such, the comma separators should be inserted appropriately. The Ansible anchors/aliases syntax are used throughout the file so that there's one only one place to update a value, thus avoiding duplicated data.

Ansible has several work directories on the RAC nodes, of which the top directory is specified in global.work_dir. Under this directory, four sub-directories are created: scripts_dir, saved_dir, done_dir, and files_dir. Briefly, scripts_dir contains scripts to perform role tasks, saved_dir contains AIX system files before they are modified, done_dir contains files that signal the completion of tasks that are non-idempotent, and files_dir contains data files needed by the scripts.

Note : For AIX 73, in ansible.cfg file update "interpreter_python = /usr/opt/freeware/bin/python3" before running the playbook

How to run the playbook

After the ansible.cfg, inventory.yml, and vars.yml has been updated and reviewed, run the playbook **at the top directory** like so:

```
$ ansible-playbook play.yml 2>&1 | tee play.out
```

You can also run single or multiple roles using tags option

```
ansible-playbook  play.yml –-tags bootstrap 2>&1 | tee play_bootstrap.out
```

```
ansible-playbook play.yml –-tags preconfig 2>&1 | tee play_preconfig.out
```

ansible-playbook play.yml –-tags config,install 2>&1 | tee play.out

Below is the content of playbook

```
$ cat play.yml
---

# play.yml

- hosts: racnodes
  gather_facts: no
  vars_files:
    - vars.yml
#    - dev_grid_oracle_ofa_vars.yml
  roles:
    - role: bootstrap
      vars:
        download_dir: "~"
        target_dir: "/tmp/.ansible.cpdir"
      tags: bootstrap
    - role: preconfig
      tags: preconfig
    - role: config
      tags: config
    - role: install
      tags: install
```

Ansible will run the four roles sequentially. The four roles are bootstrap, preconfig, config and install.

When rerunning the failing role, it is helpful to use -vvv debug option to obtain more debug information with more readable STDOUT and STDERR messages.

Suppose Ansible failed in config role, and the issue is fixed, rerun starting from config role using the --tags option like this:

```
$ ansible-playbook -vvv play.yml –-tags config,install 2>&1 | tee play.out
```

Ansible will run the config and install roles.

Note: While installing openssl dependent packages/rpms using yum or dnf it will fail if openssl version is < 1.1. Refer Appendix for more details.

RAC setup on PowerVS LPARs

1) Create LPARs that are need for RAC cluster with Networks that are needed for oracle RAC
2) Create small size LPAR, configure DNS and NFS – to stage oracle software
3) Set the root user password for RAC nodes
4) Add Local and shared storage LUNs that are needed for oracle RAC installation
5) Extend the rootvg volume group, In PowerVS LPAR the boot LUN size is fixed AIX72/71 – 20G and AIX73 - 25G. This is need to accommodate swap space and ansible remote location requirements. Here hdiskX is the newly added disk, this disk size can be > 50G.

```
chvg  -t 4 rootvg
extendvg -f rootvg hdiskX
chlv -x 1024 hd6
```

6) Fill the vars.yml file and Execute the play book

Executing Collection using Ansible Automation Platform 2 (AAP2)

Ansible Automation Platform 2 is fully restructured for a hybrid cloud-native world and enables to execute automation in containerized environments.

Here in this section we will show how to create the containerized image and execute the playbook using execution environment(Containerized image).

```
$ cat execution-environment.yml
---
version: 1

build_arg_defaults:
  EE_BASE_IMAGE: 'quay.io/ansible/ansible-runner:latest'

dependencies:
  galaxy: requirements.yml

additional_build_steps:
  append:
   - RUN pip install netaddr
   - RUN yum install expect -y

$ cat requirements.yml
---
collections:
  - ibm.power_aix
  - ansible.utils
```

At first "ansible-builder" is used to build the Container Image. For more info regarding ansible-builder refer to below https://access.redhat.com/documentation/en-us/red_hat_ansible_automation_platform/2.0-ea/html-single/ansible_builder_guide/index

- Create the execution-environment.yml

- Create container image using ansible builder

- A context directory is created, and we can see the Container file inside it. Displaying the content of Container/Docker file

```
$ cat Containerfile
ARG EE_BASE_IMAGE=quay.io/ansible/ansible-runner:latest
ARG EE_BUILDER_IMAGE=quay.io/ansible/ansible-builder:latest

FROM $EE_BASE_IMAGE as galaxy
ARG ANSIBLE_GALAXY_CLI_COLLECTION_OPTS=
USER root

ADD _build /build
WORKDIR /build

RUN ansible-galaxy role install -r requirements.yml --roles-path /usr/share/ansible/roles
RUN ansible-galaxy collection install $ANSIBLE_GALAXY_CLI_COLLECTION_OPTS -r
requirements.yml --collections-path /usr/share/ansible/collections

FROM $EE_BUILDER_IMAGE as builder

COPY --from=galaxy /usr/share/ansible /usr/share/ansible

RUN ansible-builder introspect --sanitize --write-bindep=/tmp/src/bindep.txt --write-
pip=/tmp/src/requirements.txt
RUN assemble

FROM $EE_BASE_IMAGE
USER root

COPY --from=galaxy /usr/share/ansible /usr/share/ansible

COPY --from=builder /output/ /output/
RUN /output/install-from-bindep && rm -rf /output/wheels
RUN pip install netaddr
RUN yum install expect -y
```

- ansible-navigator is used for executing the playbook in CLI using execution environments (Container image). Go to power_aix_oracle_rac_asm collection and create ansible-navigator.yml file

For more details regarding ansible-navigator refer to below link

https://access.redhat.com/documentation/en-us/red_hat_ansible_automation_platform/2.3/html/ansible_navigator_creator_guide/index

Tested host configuration A (4-node cluster)

- Four AIX 72 TL4 SP 1 LPARs were configured from an IBM Power8®processor-based server. Tested with 2-node (VIO client LPARs) cluster and 4-node cluster (2 VIO client LPARs and 2 non-VIO client LPARs). A variant of this cluster was an 8-node cluster with AIX 72 TL5 SP2 was tested too. All LPARs were non-VIO clients.
- Each LPAR has 4 CPUs and 64 GB memory. (For production, the minimum is 4 CPUs and 32 GB of memory)
- Oracle Grid and Database install has been tested with the default locale – i.e. on AIX, LANG=en_US, in Grid, NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P1.

```
$ cat ansible-navigator.yml
---
ansible-navigator:
  execution-environment:
    container-engine: podman
    enabled: True
    environment-variables:
      set:
        ANSIBLE_CONFIG: ansible.cfg
    image: oracle_rac_aix_ee:latest


$ ansible-navigator run play.yml --pp=missing -m stdout  2>&1 | tee play_aap2.out
```

- Oracle 19c Grid Infrastructure 19.11 and Database 19.11 and also with 19.12.

- Four networks on each RAC node: Internet access network, Oracle public network, and two Oracle Interconnect networks.
- Disk configuration

  The list below represents the hard disk resources in our tests for installing and running Grid and a database only install configuration. The sizes for the disks listed are over-sized for typical scenarios. Determine the sizes as appropriate. More disks are required for ASM disk groups to create databases on the cluster.

Disk	Renamed Disk	Size (GB)	Shared	Purpose
hdisk0		128	No	Root disk
hdisk1		130	No	AIX volume group for Grid HOME filesystem
hdisk2		130	No	AIX volume group for Grid HOME filesystem
hdisk23	OCRVOTE_hd1	20	Yes	ASM OCR/VOTE disk group
hdisk24	OCRVOTE_hd2	20	Yes	ASM OCR/VOTE disk group
hdisk25	OCRVOTE_hd3	20	Yes	ASM OCR/VOTE disk group
hdisk26	OCRVOTE_hd4	20	Yes	ASM OCR/VOTE disk group
hdisk27	OCRVOTE_hd5	20	Yes	ASM OCR/VOTE disk group
hdisk28	OCRVOTE_hd6	20	Yes	ASM OCR/VOTE disk group
hdisk5	GIMR_hd5	128	Yes	Grid Infrastructure Management Repository disk group (aka. MGMT)
hdisk20	ACFS_hd1	128	Yes	ACFS disk group for Oracle database HOME

hdisk21	ACFS_hd2	128	Yes	ACFS disk group for Oracle database HOME
hdisk12	ASMDATA1	128	Yes	ASM disk group named ASMDATA with NORMAL redundancy
hdisk13	ASMDATA2	128	Yes	ASM disk group named DATA1 with NORMAL redundancy
hdisk14	ASMDATA3	128	Yes	ASM disk group named DATA1 with NORMAL redundancy
hdisk15	ASMDATA4	128	Yes	ASM disk group named DATA1 with NORMAL redundancy
hdisk16		128	Yes	ASM disk group named DATA2 with EXTERNAL redundancy
hdisk17		128	Yes	ASM disk group named DATA2 with EXTERNAL redundancy
hdisk18		128	Yes	ASM disk group named DATA2 with EXTERNAL redundancy
hdisk19		128	Yes	ASM disk group named DATA2 with EXTERNAL redundancy

The following are storage suggestions to bring up a functional RAC, it is not a recommendation for production purpose. Consult Oracle 19c RAC documentation for guidance to meet the production requirements.

- The root disk should be at least 64 GB.
- The volume group for Grid HOME should have at least 70 GB.
- Total usable capacity of the OCR/VOTE disks should be 2 GB, with ASM disk group EXTERNAL redundancy. Tested with EXTERNAL and NORMAL redundancy levels.
- For ACFS disk group disks with EXTERNAL redundancy, the total capacity should be at least 30 GB (20 GB + 10 GB free space) for a 4-node cluster.
- For GIMR disks with EXTERNAL redundancy, total capacity should be at least 35 GB.

Tested Ansible controller configuration A (4-node cluster)

• Red Hat Enterprise Linux 8.2 • Architecture: x86_64 • CPU MHz: 2596.99 • CPU(s): 2 • Thread(s) per core: 1 • Memory: 4GB • Ansible 2.9.11 • Python 3.6.8	• Red Hat Enterprise Linux 8.4 • Architecture: x86_64 • CPU MHz: 2596.99 • CPU(s): 6 • Thread(s) per core: 2 • Memory: 32GB • Ansible 2.9.24 • Python 3.6.8	• Red Hat Enterprise Linux 8.4 (Linux KVM) • Architecture: x86_64 • CPU MHz: 2596.99 • CPU(s): 2 • Thread(s) per core: 2 • Memory: 3GB • Ansible 2.9.24 • Python 3.6.8

In the collection, there are sample files that can be used as templates:

- ansible.cfg

  It is not expected anything needs to be updated.

- inventory.yml

  Update racnodes.hosts list with IP addresses for the RAC nodes

- vars.yml

  Read the comments to ensure the correct values are entered.

Tested host configuration B (2-node cluster)

- Two AIX 72 TL5 SP3 (pre-released) LPARs were configured from an IBM Power9®processor-based server. The LPARs are VIO clients each with 8 CPUs and 128 GB memory.
- Oracle Grid and Database install has been tested with the default locale – i.e. on AIX, LANG=en_US, in Grid, NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P1.
- Oracle 19c Grid Infrastructure 19.8 and Database 19.8.
- Three networks on each RAC node: Combined Internet access and Oracle public network, and two Oracle Interconnect networks.
- Disk configuration

  The list below represents the hard disk resources in our tests for installing and running Grid and a database only install only configuration. The sizes for the disks listed maybe over-sized for typical scenarios. Determine the sizes as appropriate. More disks are required for ASM disk groups to create databases on the cluster.

Disk	Renamed Disk	Size (GB)	Shared	Purpose
hdisk0		200	No	Root disk
hdisk1	Rename to hdisk11 on 1st node	100	Yes	AIX volume group for Grid HOME filesystem for 1st node
hdisk2	Rename to hdisk11 on the 2nd node	100	Yes	AIX volume group for Grid HOME filesystem for 2nd node
hdisk3	ASMACFS1	100	Yes	ASM ACFS disk group
hdisk4	ASMACFS2	100	Yes	ASM ACFS disk group
hdisk5	ASMOCRVOTE1	100	Yes	ASM OCR/VOTE disk group
hdisk6	ASMOCRVOTE2	100	Yes	ASM OCR/VOTE disk group
hdisk7	ASMOCRVOTE3	100	Yes	ASM OCR/VOTE disk group
hdisk8	ASMGIMR1	100	Yes	Grid Infrastructure Management Repository disk group (aka. MGMT)

- There are no local disks available in this cluster for creating a volume group to Grid HOME, except hdisk0 (rootvg), so shared hdisk1 on the 1st node was renamed to hdisk11 and shared hdisk2 on the 2nd node was renamed to hdisk11 before Ansible playbook runs. This maneuver was to work around the limitation of the disk configuration on the cluster and to satisfy the disk names must be consistent across the cluster.

Tested Ansible controller configuration B (2-node cluster)

- Red Hat Enterprise Linux 8.4/8.5
- Architecture: x86_64

- CPU MHz: 2600
- CPU(s): 6/8
- Thread(s) per core: 2
- Memory: 32GB/16GB
- Ansible 2.9.24/ Ansible core 2.12.1
- Python 3.6.8/ 3.8.8

Other tested configurations

The other tested configurations were for testing the automation with various number of cluster nodes using Oracle 19c RAC Release 19.12. The details of 2-node and 4-node configurations are covered above. It would take up a lot of space if the details of all configurations are listed, so the they are skipped.

Appendix

- **Programming languages used**

  Apart from YAML and Jinja2, many tasks are implemented in Korn Shell 93 and Perl.

- **Customization/enhancement requests**

  The files under each role directory have been tested. Unless additional functionalities/features are desired beyond this release provides, making changes should proceed with caution. Good understanding of the task flow and dependency is important to ensure successfully outcome.

  Alternatively, it's recommended to submit enhancement requests.

- **Limitations**
  - The Oracle Grid HOME is on a local JFS2 filesystem created within an AIX volume group whereas Oracle database HOME is created on shared disks behind an ASM's ACFS disk group.
  - NFS is needed to stage the oracle software
  - Passwords in plain text are stored in in vars.yml and in the scripts. Some templated scripts contain passwords are pushed out to the RAC nodes. Security measures should be taken such as limit the access to the Ansible controller and the RAC nodes prior and during the installation and change the passwords as soon as the installation is successful.
  - Proxy environment is not supported.
  - Grid Naming Service (GNS) virtual IP address is not supported.
  - OCR and voting disks are supported on ASM.

- **Openssl check**

  All openssl dependent packages updated after Aug 5, 2022 require openssl 1.1. The bootstrap and config roles which use yum/dnf will get fail if you are trying to install the openssl dependent packages/rpms when openssl version is < 1.1.

  Openssl fileset needs to be upgraded manually, refer to below url for steps

  https://www.ibm.com/support/pages/node/6833478

  https://www.ibm.com/support/pages/node/720655

```
# lslpp -l|grep openssl
  openssl.base          1.0.2.2104  COMMITTED  Open Secure Socket Layer
  openssl.license        1.0.2.2104  COMMITTED  Open Secure Socket License
  openssl.man.en_US      1.0.2.2104  COMMITTED  Open Secure Socket Layer
  openssl.base          1.0.2.2104  COMMITTED  Open Secure Socket Layer
```

- **Disk headers and PVIDs**

  Disks may have AIX volume group, Oracle ASM diskgroup headers, and PVIDs left on the disks by a previous project. When creating a volume group or an ASM diskgroup, the PVIDs and/or headers cause the volume group or ASM diskgroup creation to fail. If it is certain the disks can be reused, clear the header on the disks before starting the playbook like this:

```
dd if=/dev/zero of=/dev/hdiskX bs=1024k count=100
```

  To determine a disk has an ASM diskgroup header, use lquerypv -h /dev/hdiskX

  https://www.ibm.com/support/pages/do-not-mix-oracle-asm-disks-lvm

  Example showing ASM header:

```
# lquerypv -h /dev/hdisk23

00000000   00820101 00000000 80000000 BF477E44  |............G~D|

00000010   00000000 00000000 00000000 00000000  |................|

00000020   4F52434C 4449534B 00000000 00000000  |ORCLDISK........|

00000030   00000000 00000000 00000000 00000000  |................|

00000040   13000000 00000103 4F435256 4F54455F  |........OCRVOTE_|

00000050   30303030 00000000 00000000 00000000  |0000............|

00000060   00000000 00000000 4F435256 4F544500  |........OCRVOTE.|

00000070   00000000 00000000 00000000 00000000  |................|

00000080   00000000 00000000 4F435256 4F54455F  |........OCRVOTE_|

00000090   30303030 00000000 00000000 00000000  |0000............|

000000A0   00000000 00000000 00000000 00000000  |................|
000000B0   00000000 00000000 00000000 00000000  |................|
000000C0   00000000 00000000 01F95E4D 8CACD800  |..........^M....|
000000D0   01F95E4D 9DE2F000 02001000 00400000  |..^M.........@..|
000000E0   0006EE80 00001400 00000003 00000001  |................|
000000F0   00000002 0000000A 00000000 00000000  |................|
```

To determine if a disk has an on-disk PVID, use lqueryvg -Ptp hdiskX

https://www.ibm.com/support/pages/resolving-missing-or-removed-disks-aix-lvm

If PVID exists, the exit code is 0, otherwise it is 1.

To determine if a disk has a Volume Group Descriptor Area (VGDA), use readvgda -t hdiskX.

https://www.ibm.com/support/pages/how-determine-volume-group-physical-partition-size-readvgda

If VGDA exists, the exit code is 0, otherwise it is 1.

Alternatively, if disks can be reused for sure, the vars.yml config.asmdisks.diskgroups and config.ora_vg has options “clear_pvids” and “zero_disks” which free the user from executing the extra commands to clear the PVIDs, AIX volume group headers, and ASM diskgroup headers.

Trademarks

IBM®is a registered trademark of IBM Corporation in the United States and other countries

POWER9®is a registered trademark of IBM Corporation in the United States and other countries

POWER8®is a registered trademark of IBM Corporation in the United States and other countries

POWER® is a registered trademark of IBM Corporation in the United States and other countries

AIX® is a registered trademark of IBM Corporation in the United States and other countries

Oracle® is a registered trademark of Oracle Corporation in the United States and other countries

Ansible®is a registered trademark of Red Hat® Inc. in the United States and other countries

Red Hat®is a registered trademark of Red Hat Inc. in the United States and other countries