

Upgrade RAC Instance GI and Databases from 12c to 19c using Ansible

**Introduction:** This readme helps to automate the upgrade of RAC Instance Grid Infrastructure along with Multiple Oracle Databases on ASM/JFS from Oracle 12c to 19c. This covers the installation of 19c RAC Instance GI and upgrade of the existing 12.x to 19c and installation of 19c RDBMS. Along with the database upgrade leveraging Oracle’s “autoupgrade.jar” tool. [About Oracle Database AutoUpgrade.](#)

Below figures provide a pictorial representation of the upgrade process using this Ansible collection.

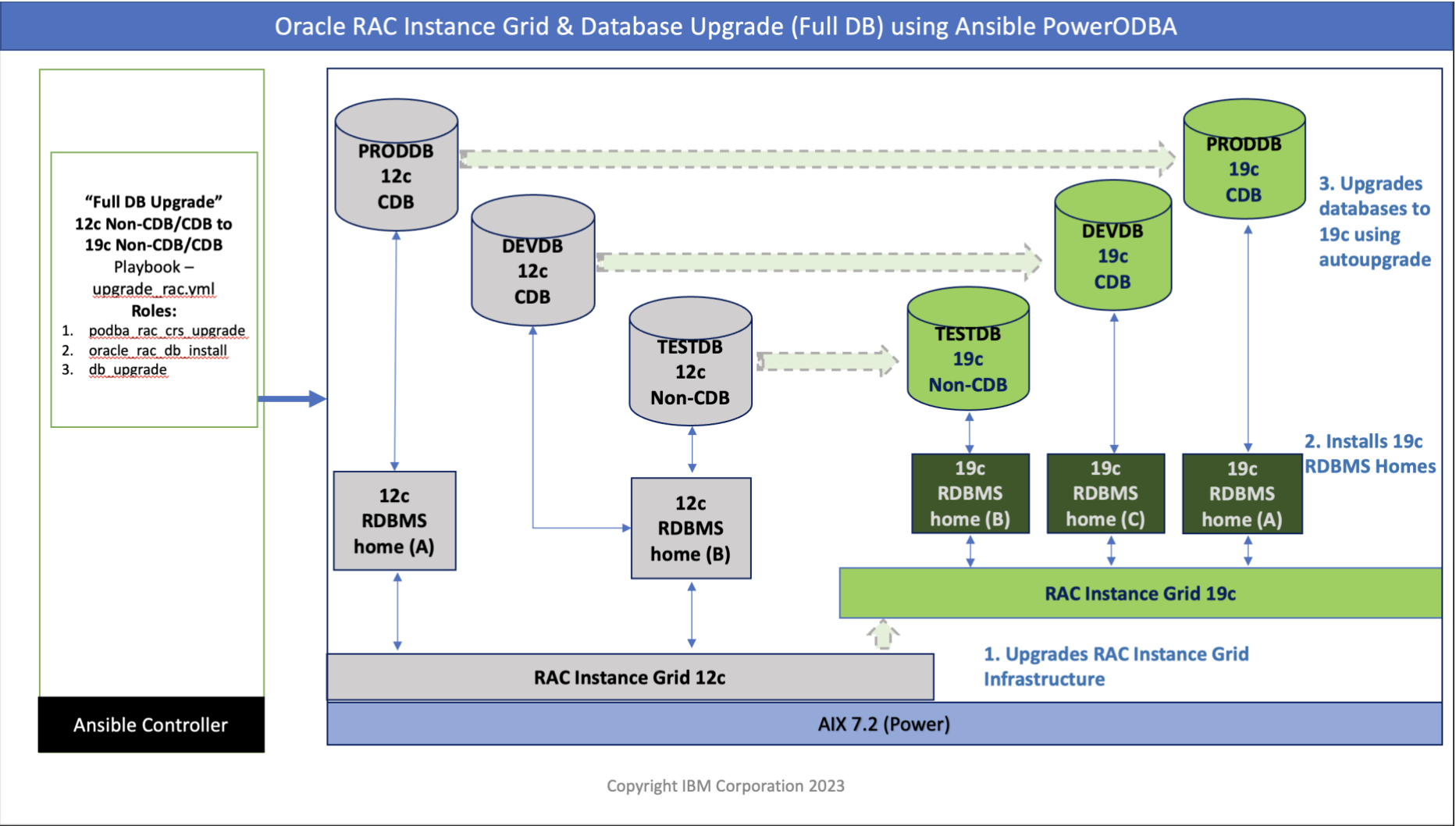


Fig: 1 Upgrading 12c Non-CDB/CDB to 19c Non-CDB/CDB

Note: All the pluggable databases inside the container database will be upgraded from 12c to 19c as part of upgrade process.

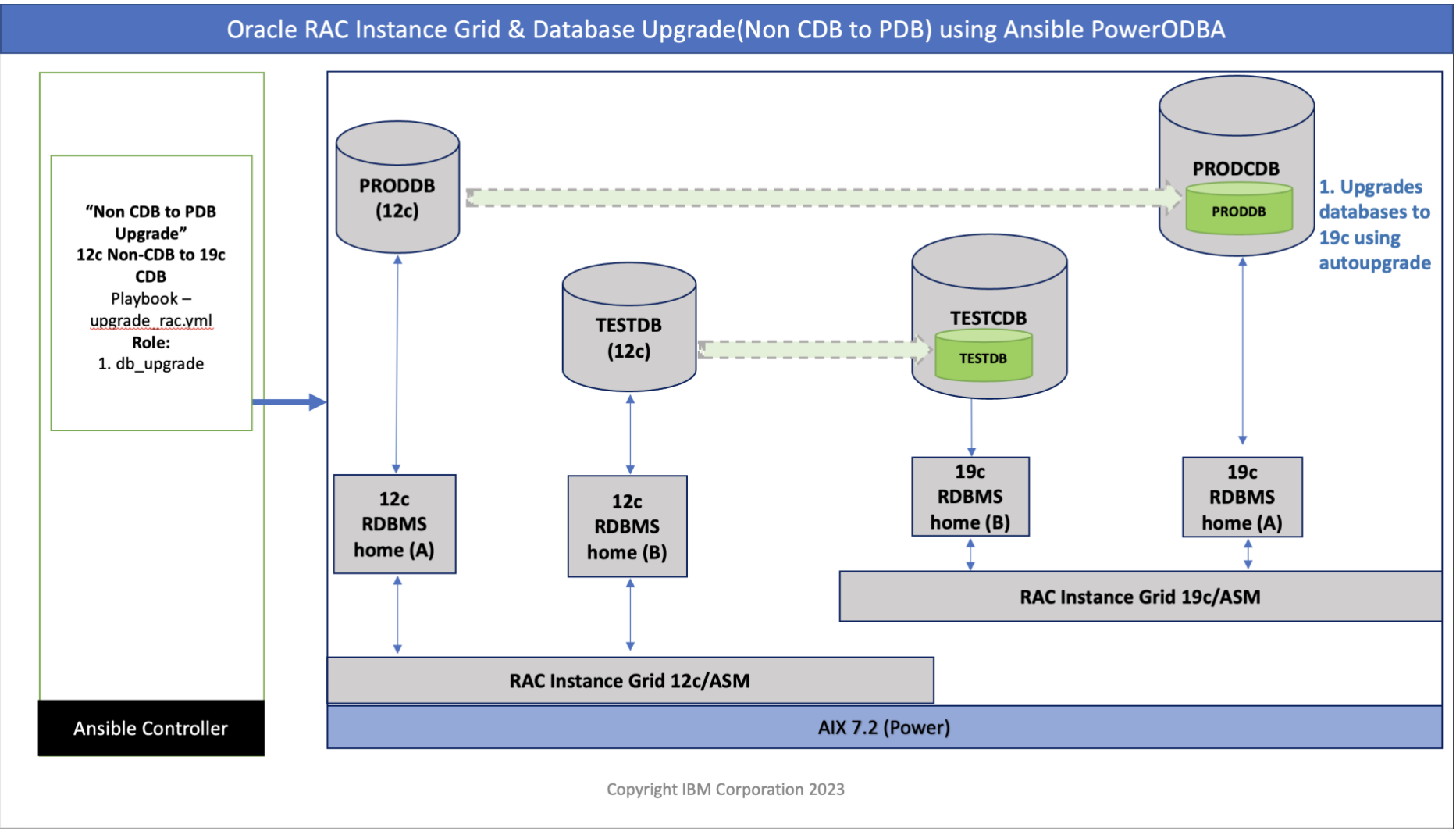


Fig: 2 Upgrading 12c Non-CDB to 19c CDB

Note: When upgrading from 12c Non CDB to 19c CDB, the 19c Container database should be created manually and the variable file should be updated before running the playbooks. Please read “Other scenarios” of “Upgrade Scenarios” provided in the later section of this readme.

## File System architecture:

1. **Path to the collection:** `$ansible-collection-install-dir/ibm/power_aix_oracle_rac_upgrade`.
2. **upgrade\_rac.yml:** This is the *playbook* file which is responsible for installation and upgrade of 19c RAC instance Grid & databases by calling the respective roles. The file is under - ***“ansible-collection-install-dir”/ibm/power\_aix\_oracle\_rac\_upgrade/playbooks***. Only the managed host’s hostname must be updated in this file.
3. **inventory.yml:** This file is provided in the collection which contain all the managed hosts details. It is NOT mandatory to use only this file, if you already have an inventory file defined in another location, that can be used also.
4. **upgrade\_rac\_vars.yml:** This file contains all the variables required to perform the upgrade. It is under is under - ***“ansible-collection-install-dir”/ibm/power\_aix\_oracle\_rac\_upgrade/playbooks/vars/upgrade***. Specification of each variable is provided in this file itself.
5. **vault.yml:** The sys user password of ASM must be mentioned in this file, this file is in ***“ansible-collection-install-dir”/ibm/power\_aix\_oracle\_rac\_upgrade/playbooks/vars***. It must be encrypted using “ansible-vault” after the password is stored in the file. Ansible Vault is a security utility provided by Ansible to encrypt files which contain sensitive information such as passwords. Refer: [A brief introduction to Ansible Vault | Enable Sysadmin \(redhat.com\)](#)  
`$ ansible-vault encrypt vault.yml`
6. **Roles:** There are four ansible roles which will be used to perform the upgrade. Description of each role is given below.

Ansible Roles to Upgrade RAC GI & DBs	
rac_crs_precheck	rac_crs_upgrade
Performs the precheck required for 19c Grid & RDBMS RAC upgrade	Upgrades RAC Instance 12c Grid to 19c with RU
<ol style="list-style-type: none"><li>1. Checks if 19c RAC Instance Grid Infrastructure is already installed.</li><li>2. Checks for source (12c) grid setup.</li><li>3. Checks if CRS release version versions are same across nodes.</li><li>4. Checks the required value of maxuproc value.</li><li>5. Checks for mandatory patch on source (12c) grid home for 19c upgrade.</li><li>6. Checks if unzip is available.</li><li>7. Checks free space in Grid home path and fails if free space is less than 90GB, it is mandatory check set by Oracle to have 80GB of free space to do the patching.</li></ol>	<ol style="list-style-type: none"><li>1. Checks for any 19c Grid software already installed.</li><li>2. Extracts the 19c Grid software.</li><li>3. Backups up OPatch and extracts the latest one.</li><li>4. Extracts Release update patch .</li><li>5. Executes rootpre.sh.</li><li>6. Creates &amp; copies grid_19c_install.rsp response file from template to the target lpar.</li><li>7. Executes gridSetup.sh (with CRS SWONLY option) in silent mode .</li><li>8. Apply patch on New 19c standalone GI Home.</li><li>9. Executes Grid home detach script.</li><li>10. Runs cluvfy.sh and exits in case of any issues are reported by the utility.</li><li>11. Creates &amp; copies grid_19c_upgrade.rsp response file from template to the target lpar.</li><li>12. Executes gridSetup.sh (with upgrade option) in silent mode from patched home. It will show the status and lists the log files upon completion of the new 19c Grid Installation.</li><li>13. Executes rootupgrade.sh.</li><li>14. Executes gridSetup.sh -silent -executeConfigTools and lists the log files upon completion.</li><li>15. Checks and displays the status of grid release patch Version.</li></ol>

Ansible Roles to Upgrade RAC GI & DBs	
oracle_rac_db_install	db_upgrade
Installs 19c Oracle database home with RU for Upgrades	Upgrades the database using autoupgrade.jar utility.
<ol style="list-style-type: none"><li>1. Checks for any 19c DB upgrade already done.</li><li>2. Checks free space in RDBMS home path and fails if free space is less than 30GB, it is mandatory check set by Oracle to have 30GB of free space to do the installation .</li><li>3. Extracts the 19c RDBMS software.</li><li>4. Backups up OPatch and extracts the latest one.</li><li>5. Extracts Release update patch.</li><li>6. Executes rootpre.sh.</li><li>7. Creates &amp; copies oracle_19c_install.rsp response file from template to the target lpar.</li><li>8. Installs 19c RDBMS software (with INSTALL_DB_SWONLY option) for Single instance database installation.</li><li>9. Executes root.sh.</li><li>10. Apply patch on New 19c standalone RDBMS Home.</li><li>11. Detach oracle database home</li><li>12. Installs 19c RDBMS software (with INSTALL_DB_SWONLY option) for RAC using patched home.</li><li>13. Executes root.sh.</li><li>14. Continues to create multiple Oracle homes based on the number of “target_db_home” variables provided in the variables file.</li></ol>	<ol style="list-style-type: none"><li>1. Creates a stage directory to place the autoupgrade configuration file.</li><li>2. Checks the DB name in oratab.</li><li>3. Checks DB is up and running or not.</li><li>4. Generates autoupgrade configuration file based on the inputs provided in the variables file.</li><li>5. Executes autoupgrade in analyze mode. [User must review the “analyze” results and fix them before running the “deploy” mode.</li><li>6. User should execute the playbook with deploy tag for autoupgrade to run in deploy mode.</li></ol>

**Important Steps to note before doing the upgrade:**

1. Prior to the upgrade process, apply the required patches on the 12c environment to avoid unintended errors. Refer **MOS Doc ID 2539751.1**.
  2. Stage the 19c Grid home software and 19c RDBMS home software, Release Update (RU) patch & latest Opatch utility zip files either in Local/NFS or on the Ansible Controller and update them in the variables file [power\_aix\_oracle\_rac\_upgrade/playbooks/vars/upgrade/upgrade\_rac\_vars.yml].
  3. Update the parameter *remote\_tmp* in the *ansible.cfg* file to a path with minimum 16GB of space. Ansible will temporarily use it for unarchiving the binaries.
  4. Always use the latest autoupgrade.jar file [AutoUpgrade Tool (Doc ID 2485457.1)] and update that path for the variable “autoupgrade\_file\_loc” in the variables file [power\_aix\_oracle\_rac\_upgrade/playbooks/vars/upgrade/upgrade\_rac\_vars.yml].
  5. The free space in the installer path of 19c homes should be more than 90GB.
  6. If you want the autoupgrade to create a restore point, enable Flashback mode, and maintain sufficient space for the Flash Recovery Area (FRA) to avoid failure during the upgrade process.
    - SQL> alter system set db\_recovery\_file\_dest\_size=35G; -- Increase the size as per your environment.
    - SQL> alter system set db\_recovery\_file\_dest='+FRA'; -- For example, the diskgroup name FRA was used. JFS path can also be used.
    - SQL> shutdown immediate;
    - SQL> startup mount;
    - SQL> alter database archivelog;
    - SQL> alter database flashback on;
    - SQL> alter database open;
  7. To proceed with the upgrade without creating a restore point, update this variable “restoration” in the “databases” section to ‘no’ in the variables file power\_aix\_oracle\_rac\_upgrade/playbooks/vars/upgrade/upgrade\_rac\_vars.yml
  8. The list of 12c databases running on ASM must be mentioned in the variable “source\_db\_name” of the “databases” section in power\_aix\_oracle\_rac\_upgrade/playbooks/vars/upgrade/upgrade\_rac\_vars.yml file. The listed databases will be shut down during RAC Instance Grid Infrastructure (rootupgrade.sh) upgrade.
  9. **This playbook won’t backup the databases or binaries.** Please use a standard backup strategy before starting the upgrade process in case of a fallback plan.
  10. Always run the playbooks in a vnc viewer to avoid ssh timeouts.
  11. Following tags are provided:
    - a. rac\_crs\_precheck: This will invoke the role “rac\_crs\_precheck”. Performs the precheck required for 19c Grid & RDBMS RAC upgrade from 12c to 19c.
    - b. podba\_rac\_crs\_upgrade: This will invoke the role “podba\_rac\_crs\_upgrade”. Upgrades RAC Grid Infrastructure to 19c from 12c.
    - c. oracle\_rac\_db\_install: This will invoke the role “oracle\_rac\_db\_install”. Installs 19c RAC Oracle Homes for database upgrades from 12c to 19c.
    - d. predbupgrade: This will invoke prerequisite part of “db\_upgrade” role which will do prechecks on the existing databases running on the lpar.
    - e. analyze: This will invoke "autoupgrade" analyze mode which is a section of “db\_upgrade” role.
    - f. deploy: This will invoke a section of “db\_upgrade” role which will invoke "autoupgrade" deploy mode, which is the core database upgrade mode.
  12. These playbooks will create three directories (podba\*) in /tmp. They should NOT be removed until the upgrade process completes otherwise it will compromise idempotency.
  13. Try this on a non-production environment first before using it on a Production environment.

**Upgrade scenarios:**

1. **Full Stack Upgrade:** Upgrade RAC Instance GI & all the databases running on ASM. Playbook must be executed twice as shown below. The first command will perform 19c RAC instance GI installation and upgrade, installation of 19c RDBMS along with Autoupgrade Analyze mode on the databases. Users must review the logs and fix any errors/recommendations reported by the autoupgrade tool and rerun the playbook with “deploy” tag.
    - 1) ansible-playbook upgrade\_rac.yml -i inventory.yml --ask-vault-pass --tags rac\_crs\_precheck,rac\_crs\_upgrade,oracle\_rac\_db\_install,predbupgrade,analyze
    - 2) ansible-playbook upgrade\_rac.yml -i inventory.yml --ask-vault-pass --tags deploy
    - 3) ansible-playbook upgrade\_rac.yml -i inventory.yml --ask-vault-pass
  2. **Other scenarios:**
    - a) To upgrade only RAC Instance Grid, run the following command.  
*\$ ansible-playbook upgrade\_rac.yml -i inventory.yml --ask-vault-pass --tags rac\_crs\_upgrade*
    - b) If the databases are running on JFS then RAC Instance Grid upgrade is not required. In such case, skip the role “rac\_crs\_upgrade”  
*\$ ansible-playbook upgrade\_rac.yml -i inventory.yml --ask-vault-pass --skip-tags rac\_crs\_upgrade*
    - c) If 19c Oracle Homes are already installed, skip the role “oracle\_install” and directly jump to “db\_upgrade”.  
*\$ ansible-playbook upgrade\_rac.yml -i inventory.yml --ask-vault-pass --tags predbupgrade,analyze*  
Review the results of analyze mode of autoupgrade and execute “deploy” mode.  
*\$ ansible-playbook upgrade\_rac.yml -i inventory.yml --ask-vault-pass --tags deploy.*
    - d) To upgrade a non-container database and plug it into a 19c Container database (upgrade & plug-in),  
*\$ ansible-playbook upgrade\_rac.yml -i inventory.yml --ask-vault-pass --tags oracle\_install*
      - Create a 19c container database. PowerODBA has a playbook “create-db.yml” to create the databases.  
*\$ ansible-playbook upgrade\_rac.yml -i inventory.yml --ask-vault-pass --tags predbupgrade,analyze*
      - Review the results of analyze mode of autoupgrade and execute “deploy” mode.  
*\$ ansible-playbook upgrade\_rac.yml -i inventory.yml --ask-vault-pass --tags deploy.*

**Playbook execution:** In the following example, we are going to upgrade Full Stack Oracle 12.1.0.2 RAC Instance GI to Oracle 19.24 along with one database. The s/w binaries and patches are placed in the local NFS. The Ansible playbook will extract it onto the target host before doing the upgrade.

**Playbook upgrade\_rac.yml:**

```
- name: Run Grid Upgrade Script on first node
  hosts: orac
  remote_user: root
  gather_facts: False
  vars_files:
    - vars/upgrade/upgrade_rac_vars.yml
    - vars/vault.yml

  roles:
    - role: podba_rac_upgrade_precheck
      tags: rac_crs_precheck
    - role: podba_rac_crs_upgrade
      tags: rac_crs_upgrade
    - role: podba_rac_db_install
      tags: oracle_rac_db_install
    - role: podba_rac_db_upgrade
      tags: db_upgrade
```

**Update the variables file:**

```
# Section A - Update the Common Variables.

ignoreprecheck: True

ora_binary_location: local
ora_nfs_host:
ora_nfs_device:
ora_nfs_filesystem:

grid_sw: /binaries/19c/V982588-01_193000_grid.zip
db_oracle_sw: /binaries/19c/V982583-01_193000_db.zip
ru_stage: /u01/19c/RU
opatch_sw: /binaries/19c/p6880880_190000_AIX64-5L.zip_opatch.zip
ru_zip: /binaries/19c/p36582629_190000_AIX64-5L_GI_RU19.24.zip
autoupgrade_util_remote:
autoupgrade_util: /u01/upgrade_121_19c/autoupgrade.jar

ora_inventory: /u01/base/oraInventory
mgmt_opt:
oms_host:
oms_port:
oms_em_user:

# Section B - Update the RAC Instance Grid Variables

grid_home_prev: /u01/app/12.1.0/grid
grid_home: /u01/19c/grid19c

gi_oracle_base: /u01/base
grid_user: oracle
grid_group: oinstall
gi_osdba_group: dba
gi_osasm_group: oinstall
gi_osoper_group:
gi_cluster_name: rac21
cluster_nodes: rac21,rac22
asm_disk_group: DATA

# Section C - Update the Variables for Oracle 19c RDBMS installation

db_oracle_base: /u01/base
db_oracle_user: oracle
db_oinstall_group: oinstall
db_osdba_group: dba
db_osoper_dba:
db_osbkup_dba:
db_osdg_dba:
db_oskm_dba:
db_osrac_dba:

# Section D - Variables for Database [Please Don't Change the dictionary list format. This format is referenced in the code.]

autoupgrade_stage: /home/oracle
global_log_dir: /u01/upgrade_121_19c/autoupgrade_1
databases:
  - source_db_name: orcl
    source_db_sid: orcl1
    source_db_home: /u01/app/oracle/product/12.1.0/dbhome_1
    target_db_home: /u01/19c/db19c
    log_dir: /u01/upgrade_121_19c/autoupgrade_1/dbupgrdlogs
    start_time: NOW
    restoration: 'yes'
    upgrade_node: localhost
    run_utlpr: 'yes'
    timezone_upg: 'yes'
    target_cdb_name:
    target_pdb_name:
```

Update the vaults.yml file with ASM sys password and encrypt it with ansible vault:

```
$ cat vars/vault.yml
asm_password: Oracle4u          # ASM sys user password for RAC Instance GI Upgrade.

$ ansible-vault encrypt vars/vault.yml
```

GI Version before the upgrade:

```
-bash-5.1$ crsctl query has releaseversion
Oracle High Availability Services release version on the local node is [12.1.0.2.0]
-bash-5.1$ hostname
rac21
```

```
[ansible@p208n149 playbooks]$ ansible-playbook upgrade_rac.yml --ask-vault-pass

PLAY [Run Grid Upgrade Script on first node]
*****

TASK [podba_rac_upgrade_precheck : Run initialization tasks]
*****
included: /home/ansible/nava/playbooks/roles/podba_rac_upgrade_precheck/tasks/init.yml for rac21, rac22

TASK [podba_rac_upgrade_precheck : Set fact init_done]
*****
ok: [rac21]
ok: [rac22]

TASK [podba_rac_upgrade_precheck : Set fact for rac_hosts from cluster_nodes]
*****
ok: [rac21]
ok: [rac22]

TASK [podba_rac_upgrade_precheck : Set fact for first_host and last_host]
*****
ok: [rac21]
ok: [rac22]

TASK [podba_rac_upgrade_precheck : Creating Temp Directory]
*****
ok: [rac22] => (item=/tmp/podba_db_upgrade/done)
ok: [rac21] => (item=/tmp/podba_db_upgrade/done)
ok: [rac21] => (item=/tmp/podba_db_upgrade/scripts)
ok: [rac22] => (item=/tmp/podba_db_upgrade/scripts)
ok: [rac22] => (item=/tmp/podba_db_upgrade/logs)
ok: [rac21] => (item=/tmp/podba_db_upgrade/logs)
ok: [rac21] => (item=/tmp/podba_gi_temp/done)
ok: [rac22] => (item=/tmp/podba_gi_temp/done)
ok: [rac21] => (item=/tmp/podba_gi_temp/scripts)
ok: [rac22] => (item=/tmp/podba_gi_temp/scripts)
ok: [rac21] => (item=/tmp/podba_gi_temp/logs)
ok: [rac22] => (item=/tmp/podba_gi_temp/logs)
ok: [rac21] => (item=/tmp/podba_install_temp/done)
ok: [rac22] => (item=/tmp/podba_install_temp/done)
ok: [rac21] => (item=/tmp/podba_install_temp/scripts)
ok: [rac22] => (item=/tmp/podba_install_temp/scripts)
ok: [rac21] => (item=/tmp/podba_install_temp/logs)
ok: [rac22] => (item=/tmp/podba_install_temp/logs)

TASK [podba_rac_upgrade_precheck : Checking if 19c GI is installed]
*****
ok: [rac21]

TASK [podba_rac_upgrade_precheck : Set fact for grid installation status]
*****
ok: [rac21]

TASK [podba_rac_upgrade_precheck : Checking if prechecks_grid script is already run]
*****
ok: [rac22]
ok: [rac21]

TASK [podba_rac_upgrade_precheck : Checking current CRS Version]
*****
ok: [rac22]
ok: [rac21]

TASK [podba_rac_upgrade_precheck : Setting fact - CRS Version]
*****
ok: [rac21]
ok: [rac22]

TASK [podba_rac_upgrade_precheck : Display crsctl query output for each node]
*****
ok: [rac21] => {
  "msg": "Node: rac21 - CRS Version: 12.1.0.2.0"
}
ok: [rac22] => {
  "msg": "Node: rac22 - CRS Version: 12.1.0.2.0"
}

TASK [podba_rac_upgrade_precheck : Fail and Exit if CRS versions do not match]
*****
skipping: [rac21]
skipping: [rac22]

TASK [podba_rac_upgrade_precheck : End play if CRS versions do not match]
*****
```

```
skipping: [rac21]

TASK [podba_rac_upgrade_precheck : Execute crsctl query on all nodes]
*****
ok: [rac22]
ok: [rac21]

TASK [podba_rac_upgrade_precheck : Setting fact - CRS Version]
*****
ok: [rac21]
ok: [rac22]

TASK [podba_rac_upgrade_precheck : Display crsctl query output for each node]
*****
ok: [rac21] => {
  "msg": "Node: rac21 - CRS Version: Oracle Clusterware release patch level is [336263397] and the complete list of patches
[19769480 20299023 20831110 21359755 21436941 21948354 22291127 22502518 22502555 ] have been applied on the local node."
}
ok: [rac22] => {
  "msg": "Node: rac22 - CRS Version: Oracle Clusterware release patch level is [336263397] and the complete list of patches
[19769480 20299023 20831110 21359755 21436941 21948354 22291127 22502518 22502555 ] have been applied on the local node."
}

TASK [podba_rac_upgrade_precheck : Check if unzip is available]
*****
ok: [rac21]
ok: [rac22]

TASK [podba_rac_upgrade_precheck : Display if unzip is available]
*****
ok: [rac21] => {
  "msg": "Unzip is available on this node."
}
ok: [rac22] => {
  "msg": "Unzip is available on this node."
}

TASK [podba_rac_upgrade_precheck : Display if unzip is not available]
*****
skipping: [rac21]
skipping: [rac22]

TASK [podba_rac_upgrade_precheck : Templating out prechecks_grid script]
*****
changed: [rac22]
changed: [rac21]

TASK [podba_rac_upgrade_precheck : Executing prechecks_grid script]
*****
changed: [rac22]
changed: [rac21]

TASK [podba_rac_upgrade_precheck : Prechecks Grid Script Output]
*****
ok: [rac21] => {
  "precheck_out.stdout_lines": [
    "Prechecks completed successfully."
  ]
}
ok: [rac22] => {
  "precheck_out.stdout_lines": [
    "Prechecks completed successfully."
  ]
}

TASK [podba_rac_upgrade_precheck : Checking freespace in GI Installation Path /u01/19c/grid19c]
*****
ok: [rac22]
ok: [rac21]

TASK [podba_rac_upgrade_precheck : Fail if insufficient freespace for /u01/19c/grid19c (90GB required)]
*****
skipping: [rac21]
skipping: [rac22]

TASK [podba_rac_crs_upgrade : Run initialization tasks]
*****
skipping: [rac21]
skipping: [rac22]

TASK [podba_rac_crs_upgrade : Checking if DB Upgrade was already done]
*****
ok: [rac21] => (item={'source_db_name': 'orcl', 'source_db_sid': 'orcl1', 'source_db_home':
'/u01/app/oracle/product/12.1.0/dbhome_1', 'target_db_home': '/u01/19c/db19c', 'log_dir':
'/u01/upgrade_121_19c/autoupgrade_1/dbupgrdlogs', 'start_time': 'NOW', 'restoration': 'yes', 'upgrade_node': 'localhost',
'run_utlpr': 'yes', 'timezone_upg': 'yes', 'target_cdb_name': None, 'target_pdb_name': None})

TASK [podba_rac_crs_upgrade : Set fact if DB Upgrade was done for any database]
*****
skipping: [rac21] => (item={'changed': False, 'stat': {'exists': False}, 'invocation': {'module_args': {'path':
'/tmp/podba_db_upgrade/done/orcl.success', 'follow': False, 'get_md5': False, 'get_checksum': True, 'get_mime': True,
'get_attributes': True, 'checksum_algorithm': 'sha1'}}, 'failed': False, 'item': {'source_db_name': 'orcl', 'source_db_sid':
'orcl1', 'source_db_home': '/u01/app/oracle/product/12.1.0/dbhome_1', 'target_db_home': '/u01/19c/db19c', 'log_dir':
'/u01/upgrade_121_19c/autoupgrade_1/dbupgrdlogs', 'start_time': 'NOW', 'restoration': 'yes', 'upgrade_node': 'localhost',
'run_utlpr': 'yes', 'timezone_upg': 'yes', 'target_cdb_name': None, 'target_pdb_name': None}, 'ansible_loop_var': 'item'})
skipping: [rac21]

TASK [podba_rac_crs_upgrade : Set fact for no upgrade done if none were found]
*****
ok: [rac21]

TASK [podba_rac_crs_upgrade : Exit if DB Upgrade was already done]
*****
skipping: [rac21]

TASK [podba_rac_crs_upgrade : End play if DB Upgrade Task was done]
*****
```



```
skipping: [rac21]

TASK [podba_rac_crs_upgrade : Checking if grid home was already upgraded]
*****
ok: [rac21]

TASK [podba_rac_crs_upgrade : Checking if prechecks_grid script is already run]
*****
ok: [rac21]
ok: [rac22]

TASK [podba_rac_crs_upgrade : Checking if new 19c home is already installed]
*****
skipping: [rac22]
ok: [rac21]

TASK [podba_rac_crs_upgrade : Checking if grid home was already upgraded]
*****
skipping: [rac22]
ok: [rac21]

TASK [podba_rac_crs_upgrade : Checking if GI S/W is extracted]
*****
skipping: [rac22]
ok: [rac21]

TASK [podba_rac_crs_upgrade : Checking if RU is extracted]
*****
skipping: [rac22]
ok: [rac21]

TASK [podba_rac_crs_upgrade : Checking if Opatch is extracted]
*****
skipping: [rac22]
ok: [rac21]

TASK [podba_rac_crs_upgrade : Creating NFS filesystem from nfshost]
*****
skipping: [rac21]
skipping: [rac22]

TASK [podba_rac_crs_upgrade : Creating GI home directory]
*****
skipping: [rac22]
ok: [rac21]

TASK [podba_rac_crs_upgrade : Creating RU Stage Directory]
*****
skipping: [rac22]
ok: [rac21]

TASK [podba_rac_crs_upgrade : Extracting GI S/W (Remote)]
*****
skipping: [rac21]
skipping: [rac22]

TASK [podba_rac_crs_upgrade : Creating the done file]
*****
skipping: [rac21]
skipping: [rac22]

TASK [podba_rac_crs_upgrade : Extracting GI S/W (Local|NFS)]
*****
skipping: [rac22]
changed: [rac21]

TASK [podba_rac_crs_upgrade : Creating the done file]
*****
skipping: [rac22]
changed: [rac21]

TASK [podba_rac_crs_upgrade : Extracting RU S/W (Remote)]
*****
skipping: [rac21]
skipping: [rac22]

TASK [podba_rac_crs_upgrade : Creating the done file]
*****
skipping: [rac21]
skipping: [rac22]

TASK [podba_rac_crs_upgrade : Extracting GI RU patch (Local|NFS)]
*****
skipping: [rac22]
ok: [rac21]

TASK [podba_rac_crs_upgrade : Creating the done file]
*****
skipping: [rac22]
changed: [rac21]

TASK [podba_rac_crs_upgrade : Backup Opatch]
*****
skipping: [rac21]
skipping: [rac22]

TASK [podba_rac_crs_upgrade : Extracting OPatch (Remote)]
*****
skipping: [rac21]
skipping: [rac22]

TASK [podba_rac_crs_upgrade : Creating the done file]
*****
skipping: [rac21]
skipping: [rac22]

TASK [podba_rac_crs_upgrade : Backup Opatch]
```

```
*****
skipping: [rac22]
changed: [rac21]

TASK [podba_rac_crs_upgrade : Extracting OPatch (Local|NFS)]
*****
skipping: [rac22]
changed: [rac21]

TASK [podba_rac_crs_upgrade : Creating the done file]
*****
skipping: [rac22]
changed: [rac21]

TASK [podba_rac_crs_upgrade : Executing rootpre.sh]
*****
skipping: [rac22]
changed: [rac21]

TASK [podba_rac_crs_upgrade : Templating out GI Response File to install CRS SWONLY]
*****
skipping: [rac22]
changed: [rac21]

TASK [podba_rac_crs_upgrade : Templating out grid_install Script]
*****
skipping: [rac22]
changed: [rac21]

TASK [podba_rac_crs_upgrade : Setting Up New 19c GI for CRS]
*****
skipping: [rac22]
changed: [rac21]

TASK [podba_rac_crs_upgrade : debug]
*****
****
skipping: [rac22]
ok: [rac21] => {
    "grid_install_out.stdout": "gridSetup.sh -executePrereqs completed successfully.
                                gridSetup.sh completed successfully."
}

TASK [podba_rac_crs_upgrade : Checking if new 19c home is already installed]
*****
skipping: [rac22]
ok: [rac21]

TASK [podba_rac_crs_upgrade : Checking if new 19c home is already patched]
*****
skipping: [rac22]
ok: [rac21]

TASK [podba_rac_crs_upgrade : Templating out grid_patch Script]
*****
skipping: [rac22]
changed: [rac21]

TASK [podba_rac_crs_upgrade : Applying patch on New 19c standalone GI home]
*****
skipping: [rac22]
changed: [rac21]

TASK [podba_rac_crs_upgrade : debug]
*****
****
skipping: [rac22]
ok: [rac21] => {
    "grid_patch_out.stdout": "RU Patch 36582629 applied successfully"
}

TASK [podba_rac_crs_upgrade : Checking if new 19c home is already patched]
*****
skipping: [rac22]
ok: [rac21]

TASK [podba_rac_crs_upgrade : Checking if grid detach was already run]
*****
skipping: [rac22]
ok: [rac21]

TASK [podba_rac_crs_upgrade : Templating out Grid home detach script]
*****
skipping: [rac22]
changed: [rac21]

TASK [podba_rac_crs_upgrade : Run Grid home detach script for each home]
*****
skipping: [rac22]
changed: [rac21]

TASK [podba_rac_crs_upgrade : Checking if grid detach was already run]
*****
skipping: [rac22]
ok: [rac21]

TASK [podba_rac_crs_upgrade : Checking if cluvfy was already run]
*****
skipping: [rac22]
ok: [rac21]

TASK [podba_rac_crs_upgrade : Templating out GI Runcluvfy]
*****
skipping: [rac22]
changed: [rac21]

TASK [podba_rac_crs_upgrade : Executing Runcluvfy script]
```



```
*****
skipping: [rac22]
changed: [rac21]

TASK [podba_rac_crs_upgrade : debug]
*****
skipping: [rac22]
ok: [rac21] => {
    "cluvfy_out.stdout": "INFO: runcluvfy.sh stage -pre crsinst -upgrade successfully completed."
}

TASK [podba_rac_crs_upgrade : Checking if cluvfy was already run]
*****
skipping: [rac22]
ok: [rac21]

TASK [podba_rac_crs_upgrade : Templating out GI Response File to upgrade]
*****
skipping: [rac22]
changed: [rac21]

TASK [podba_rac_crs_upgrade : Templating out grid_upgrade script]
*****
skipping: [rac22]
changed: [rac21]

TASK [podba_rac_crs_upgrade : Executing grid_upgrade script]
*****
skipping: [rac22]
changed: [rac21]

TASK [podba_rac_crs_upgrade : debug]
*****
skipping: [rac22]
ok: [rac21] => {
    "grid_upgrade_out.stdout": "gridSetup.sh -dryRunForUpgrade completed successfully.
                                gridSetup.sh completed successfully."
}

TASK [podba_rac_crs_upgrade : Checking if 19c GI is installed]
*****
ok: [rac21]

TASK [podba_rac_crs_upgrade : Checking if new 19c home is already patched]
*****
ok: [rac21]

TASK [podba_rac_crs_upgrade : Checking if grid home was already upgraded]
*****
ok: [rac21]

TASK [podba_rac_crs_upgrade : Checking if root Upgrade Task was already done]
*****
ok: [rac21]

TASK [podba_rac_crs_upgrade : Checking if rootupgrade.sh script was already executed on the rac21]
*****
ok: [rac21]

TASK [podba_rac_crs_upgrade : Templating out rootupgrade.sh script on all nodes]
*****
changed: [rac21]
changed: [rac22]

TASK [podba_rac_crs_upgrade : Run the rootupgrade script on the rac21]
*****
changed: [rac21]

TASK [podba_rac_crs_upgrade : Debug root_upgrade_out stdout]
*****
ok: [rac21] => {
    "root_upgrade_out.stdout": "Rootupgrade.sh completed successfully"
}

TASK [podba_rac_crs_upgrade : Checking if rootupgrade.sh script was already executed on the rac22]
*****
skipping: [rac21]
ok: [rac22]

TASK [podba_rac_crs_upgrade : Run the rootupgrade script on the rac22]
*****
skipping: [rac21]
changed: [rac22]

TASK [podba_rac_crs_upgrade : Debug root_upgrade_out stdout]
*****
skipping: [rac21]
ok: [rac22] => {
    "root_upgrade_out.stdout": "Rootupgrade.sh completed successfully"
}

TASK [podba_rac_crs_upgrade : Checking if rootupgrade.sh script was already executed]
*****
ok: [rac21]

TASK [podba_rac_crs_upgrade : Checking if rootupgrade.sh script was already executed]
*****
ok: [rac21]
ok: [rac22]

TASK [podba_rac_crs_upgrade : Post GI Upgrade Steps Templating out config_tools.sh]
*****
changed: [rac21]

TASK [podba_rac_crs_upgrade : Post GI Upgrade Steps| Executing the Config Tools script]
```

```
*****
changed: [rac21]

TASK [podba_rac_crs_upgrade : Debug config_tools_out stdout]
*****
ok: [rac21] => {
    "config_tools_out.stdout": "gridSetup.sh -executeConfigTools completed with [WARNING].
                                ACTION: Refer to the logs or contact Oracle Support Services."
}

TASK [podba_rac_crs_upgrade : Post GI Upgrade Steps | Checking if Config Tools is successful.]
*****
ok: [rac21]

TASK [podba_rac_crs_upgrade : Execute crsctl query on all nodes]
*****
ok: [rac22]
ok: [rac21]

TASK [podba_rac_crs_upgrade : Setting fact - CRS Version]
*****
ok: [rac21]
ok: [rac22]

TASK [podba_rac_crs_upgrade : Display crsctl query output for each node]
*****
ok: [rac21] => {
    "msg": "Node: rac21 - CRS Version: Oracle Clusterware release patch level is [336263397] and the complete list of patches
[19769480 20299023 20831110 21359755 21436941 21948354 22291127 22502518 22502555 ] have been applied on the local node."
}
ok: [rac22] => {
    "msg": "Node: rac22 - CRS Version: Oracle Clusterware release patch level is [336263397] and the complete list of patches
[19769480 20299023 20831110 21359755 21436941 21948354 22291127 22502518 22502555 ] have been applied on the local node."
}

TASK [podba_rac_db_install : Run initialization tasks]
*****
skipping: [rac21]
skipping: [rac22]

TASK [podba_rac_db_install : Checking if DB Upgrade was already done]
*****
ok: [rac21] => (item={'source_db_name': 'orcl', 'source_db_sid': 'orcl1', 'source_db_home':
'/u01/app/oracle/product/12.1.0/dbhome_1', 'target_db_home': '/u01/19c/db19c', 'log_dir':
'/u01/upgrade_121_19c/autoupgrade_1/dbupgrdlogs', 'start_time': 'NOW', 'restoration': 'yes', 'upgrade_node': 'localhost',
'run_utlrlp': 'yes', 'timezone_upg': 'yes', 'target_cdb_name': None, 'target_pdb_name': None})

TASK [podba_rac_db_install : Set fact if DB Upgrade was done for any database]
*****
skipping: [rac21] => (item={'changed': False, 'stat': {'exists': False}, 'invocation': {'module_args': {'path':
'/tmp/podba_db_upgrade/done/orcl.success', 'follow': False, 'get_md5': False, 'get_checksum': True, 'get_mime': True,
'get_attributes': True, 'checksum_algorithm': 'sha1'}}, 'failed': False, 'item': {'source_db_name': 'orcl', 'source_db_sid':
'orcl1', 'source_db_home': '/u01/app/oracle/product/12.1.0/dbhome_1', 'target_db_home': '/u01/19c/db19c', 'log_dir':
'/u01/upgrade_121_19c/autoupgrade_1/dbupgrdlogs', 'start_time': 'NOW', 'restoration': 'yes', 'upgrade_node': 'localhost',
'run_utlrlp': 'yes', 'timezone_upg': 'yes', 'target_cdb_name': None, 'target_pdb_name': None}, 'ansible_loop_var': 'item'})
skipping: [rac21]

TASK [podba_rac_db_install : Set fact for no upgrade done if none were found]
*****
ok: [rac21]

TASK [podba_rac_db_install : Exit if DB Upgrade was already done]
*****
skipping: [rac21]

TASK [podba_rac_db_install : End play if DB Upgrade Task was done]
*****
skipping: [rac21]

TASK [podba_rac_db_install : Checking freespace in DB Installation Path target_db_home]
*****
ok: [rac22] => (item={'source_db_name': 'orcl', 'source_db_sid': 'orcl1', 'source_db_home':
'/u01/app/oracle/product/12.1.0/dbhome_1', 'target_db_home': '/u01/19c/db19c', 'log_dir':
'/u01/upgrade_121_19c/autoupgrade_1/dbupgrdlogs', 'start_time': 'NOW', 'restoration': 'yes', 'upgrade_node': 'localhost',
'run_utlrlp': 'yes', 'timezone_upg': 'yes', 'target_cdb_name': None, 'target_pdb_name': None})
ok: [rac21] => (item={'source_db_name': 'orcl', 'source_db_sid': 'orcl1', 'source_db_home':
'/u01/app/oracle/product/12.1.0/dbhome_1', 'target_db_home': '/u01/19c/db19c', 'log_dir':
'/u01/upgrade_121_19c/autoupgrade_1/dbupgrdlogs', 'start_time': 'NOW', 'restoration': 'yes', 'upgrade_node': 'localhost',
'run_utlrlp': 'yes', 'timezone_upg': 'yes', 'target_cdb_name': None, 'target_pdb_name': None})

TASK [podba_rac_db_install : Fail if insufficient freespace for target_db_home (30GB required)]
*****
skipping: [rac21] => (item={'source_db_name': 'orcl', 'source_db_sid': 'orcl1', 'source_db_home':
'/u01/app/oracle/product/12.1.0/dbhome_1', 'target_db_home': '/u01/19c/db19c', 'log_dir':
'/u01/upgrade_121_19c/autoupgrade_1/dbupgrdlogs', 'start_time': 'NOW', 'restoration': 'yes', 'upgrade_node': 'localhost',
'run_utlrlp': 'yes', 'timezone_upg': 'yes', 'target_cdb_name': None, 'target_pdb_name': None})
skipping: [rac21]
skipping: [rac22] => (item={'source_db_name': 'orcl', 'source_db_sid': 'orcl1', 'source_db_home':
'/u01/app/oracle/product/12.1.0/dbhome_1', 'target_db_home': '/u01/19c/db19c', 'log_dir':
'/u01/upgrade_121_19c/autoupgrade_1/dbupgrdlogs', 'start_time': 'NOW', 'restoration': 'yes', 'upgrade_node': 'localhost',
'run_utlrlp': 'yes', 'timezone_upg': 'yes', 'target_cdb_name': None, 'target_pdb_name': None})
skipping: [rac22]

TASK [podba_rac_db_install : Checking if DB RU is extracted]
*****
skipping: [rac22]
ok: [rac21]

TASK [podba_rac_db_install : Creating RU Stage Directory]
*****
skipping: [rac22]
ok: [rac21]

TASK [podba_rac_db_install : Extracting RU patch (Remote)]
*****
skipping: [rac21]
skipping: [rac22]
```

```
TASK [podba_rac_db_install : Creating the done file]
*****
skipping: [rac21]
skipping: [rac22]

TASK [podba_rac_db_install : Extracting DB RU patch (Local|NFS)]
*****
skipping: [rac22]
ok: [rac21]

TASK [podba_rac_db_install : Creating the done file]
*****
skipping: [rac22]
changed: [rac21]

TASK [podba_rac_db_install : Reading Oracle Inventory File for standalone Install]
*****
ok: [rac22]
ok: [rac21]

TASK [podba_rac_db_install : Setting Fact for Inventory File]
*****
ok: [rac21]
ok: [rac22]

TASK [podba_rac_db_install : Preparing Oracle Homes List for Installation]
*****
ok: [rac21] => (item=None)
ok: [rac21]
ok: [rac22] => (item=None)
ok: [rac22]

TASK [podba_rac_db_install : Checking if Oracle home was already installed]
*****
skipping: [rac22] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
skipping: [rac22]
ok: [rac21] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})

TASK [podba_rac_db_install : Set facts for Oracle home installation]
*****
skipping: [rac22] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
skipping: [rac22]
ok: [rac21] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})

TASK [podba_rac_db_install : Checking if DB RU is extracted]
*****
skipping: [rac22]
ok: [rac21]

TASK [podba_rac_db_install : Checking if DB S/W is extracted]
*****
skipping: [rac22] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
skipping: [rac22]
ok: [rac21] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})

TASK [podba_rac_db_install : Checking if DB OPatch is extracted]
*****
skipping: [rac22] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
skipping: [rac22]
ok: [rac21] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})

TASK [podba_rac_db_install : Check if OPatch.bkup exists]
*****
skipping: [rac22] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
skipping: [rac22]
ok: [rac21] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})

TASK [podba_rac_db_install : Creating DB Home Directory]
*****
skipping: [rac22] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
skipping: [rac22]
ok: [rac21] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})

TASK [podba_rac_db_install : Extracting DB S/W (Remote)]
*****
skipping: [rac22] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
skipping: [rac22]
skipping: [rac21] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
skipping: [rac21]

TASK [podba_rac_db_install : Extracting DB S/W (Local|NFS)]
*****
skipping: [rac22] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
skipping: [rac22]
changed: [rac21] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})

TASK [podba_rac_db_install : Creating the done file]
*****
skipping: [rac22] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
skipping: [rac22]
changed: [rac21] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})

TASK [podba_rac_db_install : Backup OPatch if not already backed up]
*****
skipping: [rac22] => (item=/u01/19c/db19c)
skipping: [rac22]
changed: [rac21] => (item=/u01/19c/db19c)

TASK [podba_rac_db_install : Extracting Opatch (Remote)]
*****
skipping: [rac22] => (item=/u01/19c/db19c)
skipping: [rac22]
skipping: [rac21] => (item=/u01/19c/db19c)
skipping: [rac21]

TASK [podba_rac_db_install : Extracting Opatch (Local|NFS)]
```

```
*****
skipping: [rac22] => (item=/u01/19c/db19c)
skipping: [rac22]
changed: [rac21] => (item=/u01/19c/db19c)

TASK [podba_rac_db_install : Creating done file]
*****
skipping: [rac22] => (item=/u01/19c/db19c)
skipping: [rac22]
changed: [rac21] => (item=/u01/19c/db19c)

TASK [podba_rac_db_install : Checking if new 19c home is already installed and patched]
*****
skipping: [rac22] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
skipping: [rac22]
ok: [rac21] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})

TASK [podba_rac_db_install : Set facts for rootpre.sh scripts]
*****
skipping: [rac22] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
skipping: [rac22]
ok: [rac21] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})

TASK [podba_rac_db_install : Executing rootpre.sh]
*****
skipping: [rac22] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
skipping: [rac22]
changed: [rac21] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})

TASK [podba_rac_db_install : Copying Oracle RDBMS Install response file]
*****
skipping: [rac22] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
skipping: [rac22]
changed: [rac21] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})

TASK [podba_rac_db_install : Copying oracle_install.sh]
*****
skipping: [rac22] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
skipping: [rac22]
changed: [rac21] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})

TASK [podba_rac_db_install : Installing 19c RDBMS and applying patch]
*****
skipping: [rac22] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
skipping: [rac22]
changed: [rac21] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})

TASK [podba_rac_db_install : Oracle Install Output]
*****
ok: [rac21] => (item=['Oracle install prerequisite checks completed successfully.', 'Oracle software installation completed successfully.', 'RU Patch 36582629 applied successfully.']) => {
  "msg": [
    "Oracle install prerequisite checks completed successfully.",
    "Oracle software installation completed successfully.",
    "RU Patch 36582629 applied successfully."
  ]
}

TASK [podba_rac_db_install : Reading Oracle Inventory File Again for root.sh]
*****
ok: [rac22]
ok: [rac21]

TASK [podba_rac_db_install : Setting Fact for Inventory File after Install for root.sh]
*****
ok: [rac21]
ok: [rac22]

TASK [podba_rac_db_install : Preparing Oracle Homes List for Installation after install for root.sh]
*****
ok: [rac21] => (item=None)
ok: [rac21]
ok: [rac22] => (item=None)
ok: [rac22]

TASK [podba_rac_db_install : Check if the root.sh has been already executed]
*****
skipping: [rac22] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
skipping: [rac22]
ok: [rac21] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})

TASK [podba_rac_db_install : Set facts for root.sh scripts]
*****
skipping: [rac22] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
skipping: [rac22]
ok: [rac21] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})

TASK [podba_rac_db_install : Checking if Oracle home was already installed]
*****
skipping: [rac22] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
skipping: [rac22]
ok: [rac21] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})

TASK [podba_rac_db_install : Set facts for Oracle home installation]
*****
skipping: [rac22] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
skipping: [rac22]
ok: [rac21] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})

TASK [podba_rac_db_install : Create the flag file if root.sh does not executed]
*****
skipping: [rac22] => (item=u0119cdb19c)
skipping: [rac22]
changed: [rac21] => (item=u0119cdb19c)

TASK [podba_rac_db_install : Executing root.sh]
*****
```

```
skipping: [rac22] => (item=u0119cdb19c)
skipping: [rac22]
changed: [rac21] => (item=u0119cdb19c)

TASK [podba_rac_db_install : Reading Oracle Inventory File for RAC Install]
*****
ok: [rac21]
ok: [rac22]

TASK [podba_rac_db_install : Setting Fact for Inventory File for RAC Install]
*****
ok: [rac21]
ok: [rac22]

TASK [podba_rac_db_install : Preparing Oracle Homes List for Installation before detach home for RAC Install]
*****
ok: [rac21] => (item=None)
ok: [rac21]
ok: [rac22] => (item=None)
ok: [rac22]

TASK [podba_rac_db_install : Checking if Oracle home detach script already executed]
*****
skipping: [rac22] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
skipping: [rac22]
ok: [rac21] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})

TASK [podba_rac_db_install : Set facts for Oracle home detach scripts]
*****
skipping: [rac22] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
skipping: [rac22]
ok: [rac21] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})

TASK [podba_rac_db_install : Checking if Oracle rac home was installed successfully]
*****
skipping: [rac22] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
skipping: [rac22]
ok: [rac21] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})

TASK [podba_rac_db_install : Set facts for Oracle home installation]
*****
skipping: [rac22] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
skipping: [rac22]
ok: [rac21] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})

TASK [podba_rac_db_install : Templating out Oracle home detach script]
*****
skipping: [rac22]
changed: [rac21]

TASK [podba_rac_db_install : Run Oracle home detach script for each home]
*****
skipping: [rac22] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
skipping: [rac22]
changed: [rac21] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})

TASK [podba_rac_db_install : Reading Oracle Inventory File Again for RAC Install after detach]
*****
ok: [rac22]
ok: [rac21]

TASK [podba_rac_db_install : Setting Fact for Inventory File after detach home for RAC Install]
*****
ok: [rac21]
ok: [rac22]

TASK [podba_rac_db_install : Preparing Oracle Homes List for Installation after detach home for RAC Install]
*****
ok: [rac21] => (item=None)
ok: [rac21]
ok: [rac22] => (item=None)
ok: [rac22]

TASK [podba_rac_db_install : Checking if Oracle home detach script executed successfully]
*****
skipping: [rac22] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
skipping: [rac22]
ok: [rac21] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})

TASK [podba_rac_db_install : Set facts for Oracle home detach scripts]
*****
skipping: [rac22] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
skipping: [rac22]
ok: [rac21] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})

TASK [podba_rac_db_install : Checking if new 19c home is already installed]
*****
skipping: [rac22] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
skipping: [rac22]
ok: [rac21] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})

TASK [podba_rac_db_install : Set facts for rootpre.sh scripts]
*****
skipping: [rac22] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
skipping: [rac22]
ok: [rac21] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})

TASK [podba_rac_db_install : Executing rootpre.sh]
*****
skipping: [rac22] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
skipping: [rac22]
changed: [rac21] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})

TASK [podba_rac_db_install : Templating out Oracle RDBMS Install response file]
*****
skipping: [rac22] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
skipping: [rac22]
```

```
changed: [rac21] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})

TASK [podba_rac_db_install : Templating out oracle_racdb_install.sh script]
*****
skipping: [rac22] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
skipping: [rac22]
changed: [rac21] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})

TASK [podba_rac_db_install : Installing 19c RDBMS home in all RAC nodes]
*****
skipping: [rac22] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
skipping: [rac22]
changed: [rac21] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})

TASK [podba_rac_db_install : Oracle Install Output]
*****
skipping: [rac22] => (item=No output)
skipping: [rac22]
ok: [rac21] => (item=['Oracle install done']) => {
  "msg": [
    "Oracle install done"
  ]
}

TASK [podba_rac_db_install : Reading Oracle Inventory File Again for root.sh]
*****
ok: [rac22]
ok: [rac21]

TASK [podba_rac_db_install : Setting Fact for Inventory File after Install for root.sh]
*****
ok: [rac21]
ok: [rac22]

TASK [podba_rac_db_install : Preparing Oracle Homes List for Installation after install for root.sh]
*****
ok: [rac21] => (item=None)
ok: [rac21]
ok: [rac22] => (item=None)
ok: [rac22]

TASK [podba_rac_db_install : Checking if new 19c RAC home is installed successfully]
*****
ok: [rac22 -> rac21(129.40.76.242)] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
ok: [rac21] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
ok: [rac21] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
ok: [rac22 -> rac21(129.40.76.242)] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})

TASK [podba_rac_db_install : Set facts for 19c RAC home is installed]
*****
ok: [rac21 -> rac22(129.40.76.243)]
ok: [rac22]

TASK [podba_rac_db_install : Create the flag file for 19c RAC home install in remote node]
*****
skipping: [rac21] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
skipping: [rac21] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
skipping: [rac21]
changed: [rac22] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
changed: [rac22] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})

TASK [podba_rac_db_install : Reading Oracle Inventory File Again for root.sh]
*****
ok: [rac22]
ok: [rac21]

TASK [podba_rac_db_install : Setting Fact for Inventory File after Install for root.sh]
*****
ok: [rac21]
ok: [rac22]

TASK [podba_rac_db_install : Preparing Oracle Homes List for Installation after install for root.sh]
*****
ok: [rac21] => (item=None)
ok: [rac21]
ok: [rac22] => (item=None)
ok: [rac22]

TASK [podba_rac_db_install : Check if the root.sh has been already executed]
*****
ok: [rac22] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
ok: [rac21] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
ok: [rac22] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
ok: [rac21] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
ok: [rac22] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
ok: [rac21] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})

TASK [podba_rac_db_install : Set facts for root.sh scripts]
*****
ok: [rac21] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
ok: [rac22] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
ok: [rac21] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
ok: [rac22] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
ok: [rac21] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
ok: [rac22] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})

TASK [podba_rac_db_install : Checking if new 19c RAC home is installed successfully]
*****
ok: [rac21] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
ok: [rac22] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
ok: [rac21] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
ok: [rac22] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
ok: [rac21] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
ok: [rac22] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})

TASK [podba_rac_db_install : Set facts for 19c RAC home is installed]
*****
```

```
ok: [rac21] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
ok: [rac22] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
ok: [rac21] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
ok: [rac22] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
ok: [rac21] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
ok: [rac22] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})

TASK [podba_rac_db_install : Create the flag file if root.sh if not executed]
*****
changed: [rac21] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
changed: [rac22] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
changed: [rac21] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
changed: [rac22] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
changed: [rac21] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
changed: [rac22] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})

TASK [podba_rac_db_install : Executing root.sh across all nodes]
*****
changed: [rac22] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
changed: [rac21] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
changed: [rac22] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
changed: [rac21] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
changed: [rac22] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
changed: [rac21] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})

TASK [podba_rac_db_install : Checking if new 19c RAC home is installed successfully]
*****
ok: [rac21] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
ok: [rac21] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})
ok: [rac21] => (item={'oh': 'u0119cdb19c', 'oracle_home': '/u01/19c/db19c'})

TASK [podba_rac_db_install : Set fact for missing root done files]
*****
ok: [rac21]

TASK [podba_rac_db_install : Fail if any db root script not executed]
*****
skipping: [rac21]

TASK [podba_rac_db_install : End play if any db root script not executed]
*****
skipping: [rac21]

TASK [podba_rac_db_upgrade : Run initialization tasks]
*****
skipping: [rac21]
skipping: [rac22]

TASK [podba_rac_db_upgrade : Checking if DB Upgrade was already done]
*****
ok: [rac21] => (item={'source_db_name': 'orcl', 'source_db_sid': 'orcl1', 'source_db_home':
'/u01/app/oracle/product/12.1.0/dbhome_1', 'target_db_home': '/u01/19c/db19c', 'log_dir':
'/u01/upgrade_121_19c/autoupgrade_1/dbupgrdlogs', 'start_time': 'NOW', 'restoration': 'yes', 'upgrade_node': 'localhost',
'run_utlpr': 'yes', 'timezone_upg': 'yes', 'target_cdb_name': None, 'target_pdb_name': None})

TASK [podba_rac_db_upgrade : Set fact if DB Upgrade was done for any database]
*****
skipping: [rac21] => (item={'changed': False, 'stat': {'exists': False}, 'invocation': {'module_args': {'path':
'/tmp/podba_db_upgrade/done/orcl.success', 'follow': False, 'get_md5': False, 'get_checksum': True, 'get_mime': True,
'get_attributes': True, 'checksum_algorithm': 'sha1'}}, 'failed': False, 'item': {'source_db_name': 'orcl', 'source_db_sid':
'orcl1', 'source_db_home': '/u01/app/oracle/product/12.1.0/dbhome_1', 'target_db_home': '/u01/19c/db19c', 'log_dir':
'/u01/upgrade_121_19c/autoupgrade_1/dbupgrdlogs', 'start_time': 'NOW', 'restoration': 'yes', 'upgrade_node': 'localhost',
'run_utlpr': 'yes', 'timezone_upg': 'yes', 'target_cdb_name': None, 'target_pdb_name': None}, 'ansible_loop_var': 'item'})
skipping: [rac21]

TASK [podba_rac_db_upgrade : Set fact for no upgrade done if none were found]
*****
ok: [rac21]

TASK [podba_rac_db_upgrade : Exit if DB Upgrade was already done]
*****
skipping: [rac21]

TASK [podba_rac_db_upgrade : End play if DB Upgrade Task was done]
*****
skipping: [rac21]

TASK [podba_rac_db_upgrade : Stage Autoupgrade Utility]
*****
skipping: [rac21]
skipping: [rac22]

TASK [podba_rac_db_upgrade : Autoupgrade Full DB | Checking source database name in /etc/oratab file]
*****
skipping: [rac22] => (item=None)
skipping: [rac22]
changed: [rac21] => (item=None)
changed: [rac21]

TASK [podba_rac_db_upgrade : Fail if source database name doesn't exist in oratab file.]
*****
skipping: [rac22] => (item=None)
skipping: [rac22]
skipping: [rac21] => (item=None)
skipping: [rac21]

TASK [podba_rac_db_upgrade : Autoupgrade Full DB | Checking the status of the Source Database]
*****
skipping: [rac22] => (item=None)
skipping: [rac22]
changed: [rac21] => (item=None)
changed: [rac21]

TASK [podba_rac_db_upgrade : Autoupgrade Full DB | Fail if Source database is not running.]
*****
skipping: [rac22] => (item=None)
skipping: [rac22]
```



```
skipping: [rac21] => (item=None)
skipping: [rac21]

TASK [podba_rac_db_upgrade : Autoupgrade Non-CDB to PDB | Checking target database name in /etc/oratab file]
*****
skipping: [rac22] => (item=None)
skipping: [rac22]
changed: [rac21] => (item=None)
changed: [rac21]

TASK [podba_rac_db_upgrade : Autoupgrade Non-CDB to PDB | Fail if Target database name doesn't exist in oratab file.]
*****
skipping: [rac22] => (item=None)
skipping: [rac22]
skipping: [rac21] => (item=None)
skipping: [rac21]

TASK [podba_rac_db_upgrade : Autoupgrade Non-CDB to PDB | Checking the status of the Target Container Database]
*****
skipping: [rac22] => (item=None)
skipping: [rac22]
changed: [rac21] => (item=None)
changed: [rac21]

TASK [podba_rac_db_upgrade : Autoupgrade Non-CDB to PDB | Fail if Target database is not up.]
*****
skipping: [rac22] => (item=None)
skipping: [rac22]
skipping: [rac21] => (item=None)
skipping: [rac21]

TASK [podba_rac_db_upgrade : Autoupgrade Full DB | Generating response file for autoupgrade]
*****
skipping: [rac22] => (item=None)
skipping: [rac22]
changed: [rac21] => (item=None)
changed: [rac21]

TASK [podba_rac_db_upgrade : Autoupgrade - Analyze]
*****
skipping: [rac22] => (item={'source_db_name': 'orcl', 'source_db_sid': 'orcl1', 'source_db_home':
'/u01/app/oracle/product/12.1.0/dbhome_1', 'target_db_home': '/u01/19c/db19c', 'log_dir':
'/u01/upgrade_121_19c/autoupgrade_1/dbupgrdlogs', 'start_time': 'NOW', 'restoration': 'yes', 'upgrade_node': 'localhost',
'run_utlrlp': 'yes', 'timezone_upg': 'yes', 'target_cdb_name': None, 'target_pdb_name': None})
skipping: [rac22]
changed: [rac21] => (item={'source_db_name': 'orcl', 'source_db_sid': 'orcl1', 'source_db_home':
'/u01/app/oracle/product/12.1.0/dbhome_1', 'target_db_home': '/u01/19c/db19c', 'log_dir':
'/u01/upgrade_121_19c/autoupgrade_1/dbupgrdlogs', 'start_time': 'NOW', 'restoration': 'yes', 'upgrade_node': 'localhost',
'run_utlrlp': 'yes', 'timezone_upg': 'yes', 'target_cdb_name': None, 'target_pdb_name': None})

TASK [podba_rac_db_upgrade : debug]
*****
*****
ok: [rac21] => {
  "analyze.results[0].stdout_lines": [
    "AutoUpgrade 24.4.240426 launched with default internal options",
    "Processing config file ...",
    "+-----+",
    "| Starting AutoUpgrade execution |",
    "+-----+",
    "1 Non-CDB(s) will be analyzed",
    "Job 100 database orcl1",
    "Job 100 completed",
    "----- Final Summary -----",
    "Number of databases          [ 1 ]",
    "",
    "Jobs finished                 [1]",
    "Jobs failed                   [0]",
    "",
    "Please check the summary report at:",
    "/u01/upgrade_121_19c/autoupgrade_1/cfgtoollogs/upgrade/auto/status/status.html",
    "/u01/upgrade_121_19c/autoupgrade_1/cfgtoollogs/upgrade/auto/status/status.log"
  ]
}
skipping: [rac22]

TASK [podba_rac_db_upgrade : Autoupgrade - Deploy] *****
skipping: [rac22] => (item={'source_db_name': 'orcl', 'source_db_sid': 'orcl1', 'source_db_home':
'/u01/app/oracle/product/12.1.0/dbhome_1', 'target_db_home': '/u01/19c/db19c', 'log_dir':
'/u01/upgrade_121_19c/autoupgrade_1/dbupgrdlogs', 'start_time': 'NOW', 'restoration': 'yes', 'upgrade_node': 'localhost',
'run_utlrlp': 'yes', 'timezone_upg': 'yes', 'target_cdb_name': None, 'target_pdb_name': None})
skipping: [rac22]
changed: [rac21] => (item={'source_db_name': 'orcl', 'source_db_sid': 'orcl1', 'source_db_home':
'/u01/app/oracle/product/12.1.0/dbhome_1', 'target_db_home': '/u01/19c/db19c', 'log_dir':
'/u01/upgrade_121_19c/autoupgrade_1/dbupgrdlogs', 'start_time': 'NOW', 'restoration': 'yes', 'upgrade_node': 'localhost',
'run_utlrlp': 'yes', 'timezone_upg': 'yes', 'target_cdb_name': None, 'target_pdb_name': None})

TASK [podba_rac_db_upgrade : Copying DB upgrade file] *****
skipping: [rac22] => (item={'source_db_name': 'orcl', 'source_db_sid': 'orcl1', 'source_db_home':
'/u01/app/oracle/product/12.1.0/dbhome_1', 'target_db_home': '/u01/19c/db19c', 'log_dir':
'/u01/upgrade_121_19c/autoupgrade_1/dbupgrdlogs', 'start_time': 'NOW', 'restoration': 'yes', 'upgrade_node': 'localhost',
'run_utlrlp': 'yes', 'timezone_upg': 'yes', 'target_cdb_name': None, 'target_pdb_name': None})
skipping: [rac22]
changed: [rac21] => (item={'source_db_name': 'orcl', 'source_db_sid': 'orcl1', 'source_db_home':
'/u01/app/oracle/product/12.1.0/dbhome_1', 'target_db_home': '/u01/19c/db19c', 'log_dir':
'/u01/upgrade_121_19c/autoupgrade_1/dbupgrdlogs', 'start_time': 'NOW', 'restoration': 'yes', 'upgrade_node': 'localhost',
'run_utlrlp': 'yes', 'timezone_upg': 'yes', 'target_cdb_name': None, 'target_pdb_name': None})

TASK [podba_rac_db_upgrade : Get "Stages Summary"] *****
skipping: [rac22]
changed: [rac21]

TASK [podba_rac_db_upgrade : Display the Summary] *****
ok: [rac21] => {
  "msg": [
    "2024-11-06 11:14:22.197 INFO -----Stages Summary----- - Utilities.writeStageSummary#960 ",
    "2024-11-06 11:14:22.197 INFO          SETUP                <1 min - Utilities.writeStageSummary#960 ",
    "2024-11-06 11:14:22.198 INFO        PRECHECKS             <1 min - Utilities.writeStageSummary#960 ",
```

```
"2024-11-06 11:14:22.198 INFO DISPATCH <1 min - Utilities.writeStageSummary#960 ",
"2024-11-06 11:14:22.198 INFO COMPLETED <1 min - Utilities.writeStageSummary#960 ",
"2024-11-06 11:14:22.198 INFO End of dispatcher instance for orcl - CommonBackBone.finalJobLogging#423 ",
"2024-11-06 12:51:54.015 INFO -----Stages Summary----- - Utilities.writeStageSummary#960 ",
"2024-11-06 12:51:54.015 INFO SETUP <1 min - Utilities.writeStageSummary#960 ",
"2024-11-06 12:51:54.015 INFO GRP <1 min - Utilities.writeStageSummary#960 ",
"2024-11-06 12:51:54.015 INFO PREUPGRADE <1 min - Utilities.writeStageSummary#960 ",
"2024-11-06 12:51:54.016 INFO PRECHECKS <1 min - Utilities.writeStageSummary#960 ",
"2024-11-06 12:51:54.016 INFO PREFIXUPS <1 min - Utilities.writeStageSummary#960 ",
"2024-11-06 12:51:54.016 INFO DRAIN 7 min - Utilities.writeStageSummary#960 ",
"2024-11-06 12:51:54.016 INFO DBUPGRADE 75 min - Utilities.writeStageSummary#960 ",
"2024-11-06 12:51:54.016 INFO DISPATCH <1 min - Utilities.writeStageSummary#960 ",
"2024-11-06 12:51:54.016 INFO POSTCHECKS <1 min - Utilities.writeStageSummary#960 ",
"2024-11-06 12:51:54.017 INFO DISPATCH <1 min - Utilities.writeStageSummary#960 ",
"2024-11-06 12:51:54.017 INFO POSTFIXUPS 11 min - Utilities.writeStageSummary#960 ",
"2024-11-06 12:51:54.017 INFO POSTUPGRADE <1 min - Utilities.writeStageSummary#960 ",
"2024-11-06 12:51:54.017 INFO SYSUPDATES 1 min - Utilities.writeStageSummary#960 ",
"2024-11-06 12:51:54.017 INFO COMPLETED <1 min - Utilities.writeStageSummary#960 ",
"2024-11-06 12:51:54.018 INFO End of dispatcher instance for orcl - CommonBackBone.finalJobLogging#423 "
]
}
skipping: [rac22]

TASK [podba_rac_db_upgrade : Checking the status of the DBs] *****
skipping: [rac22] => (item=None)
skipping: [rac22]
changed: [rac21] => (item=None)
changed: [rac21]

TASK [podba_rac_db_upgrade : Status of the DBs] *****
ok: [rac21] => (item=orcl) => {
  "msg": " Status: orcl Instance orcl1 is running on node rac21\nInstance orcl2 is running on node rac22"
}
skipping: [rac22] => (item=orcl)
skipping: [rac22]

PLAY RECAP *****
rac21 : ok=141 changed=65 unreachable=0 failed=0 skipped=52 rescued=0 ignored=0
rac22 : ok=73 changed=12 unreachable=0 failed=0 skipped=172 rescued=0 ignored=0
```

GI Version after the upgrade:

```
-bash-5.1$ crsctl query has releaseversion
Oracle High Availability Services release version on the local node is [19.0.0.0.0]
-bash-5.1$ hostname
rac21
```

To execute this playbook from AAP2 use the playbook upgrade\_rac\_aap2.yml. An example document is provided here:  
[https://github.com/IBM/ansible-power-aix-oracle-dba/blob/main/docs/PowerODBA\\_using\\_AAP2.pdf](https://github.com/IBM/ansible-power-aix-oracle-dba/blob/main/docs/PowerODBA_using_AAP2.pdf)