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

Introduction: This readme helps to automate the upgrade of Single Instance Grid Infrastructure along with Multiple Oracle Databases on ASM/JFS from Oracle 12c to 19c. This covers the installation of 19c Single 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.

Collection architecture:

1. **Path to the collection**: $*ansible-collection-install-dir*/ibm/power_aix_oracle_dba.
2. **upgrade_si.yml**: This is the *playbook* file which is responsible for installation and upgrade of 19c Single instance Grid & databases by calling the respective roles. The file is under - ***"ansible-collection-install-dir"*/ibm/power_aix_oracle_dba/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_si_vars.yml**: This file contains all the variables required to perform the upgrade. It is under is under - ***"ansible-collection-installdir"*/ibm/power_aix_oracle_dba/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-installdir"*/ibm/power_aix_oracle_dba/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 three ansible roles which will be used to perform the upgrade. Description of each role is given below.

Ansible Roles to Upgrade SI GI & DBs		
si_has_upgrade	**oracle_install**	**db_upgrade**
Upgrades Single Instance 12c Grid to 19c with (or) without RU.	Installs 19c Oracle database home with (or) without RU for DB upgrades.	Upgrades the database using autoupgrade.jar utility.
1. Checks if 19c Single Instance Grid Infrastructure is already installed. 2. Checks for source (12c) grid setup. 3. Checks the required value of maxuproc value. 4. Checks for mandatory patch on source (12c) grid home for 19c upgrade. 5. Extracts the 19c Grid software. 6. Backups up OPatch and extracts latest OPatch when apply_ru is set to True. 7. Extracts Release update patch when apply_ru is set to True. 8. Checks freespace in Grid home path and fails if freespace is less than 80GB, it is mandatory check set by Oracle to have 80GB of freespace to do the patching. 9. Runs cluvfy.sh and exits in case of any issues are reported by the utility. 10. Executes rootpre.sh 11. Creates & copies grid_upgrade.rsp response file from template to the target lpar. 12. Executes gridSetup.sh (with upgrade option) in silent mode when apply_ru is set to True, it will path the binaries with latest provided Release Update patch. It will show the status and lists the log files upon completion of the new 19c Grid Installation. 13. **Upgrade:** Stops the databases on 12c DB homes which are provided in the variables file upgrade_si_vars.yml. 14. Executes rootupgrade.sh. 15. Executes gridSetup.sh -silent -executeConfigTools and lists the log files upon completion. 16. Checks and displays the status of grid services. 17. Starts the databases from 12c DB home which were stopped before the upgrade.	1. Checks for any failed 19c RDBMS installations. 2. Extracts the 19c RDBMS software. 3. Backups up OPatch and extracts the latest one when apply_ru is set to true. 4. Extracts Release update patch when apply_ru is set to True. 5. Executes rootpre.sh. 6. Creates & copies oracle_install.rsp response file from template to the target lpar. 7. Installs 19c RDBMS software. 8. Executes root.sh. 9. Continues to create multiple Oracle homes based on the number of "target_db_home" variables provided in the variables file.	1. Creates a stage directory to place the autoupgrade configuration file. 2. Checks the DB name in oratab. 3. Checks DB is up and running or not. 4. Generates autoupgrade configuration file based on the inputs provided in the variables file. 5. Executes autoupgrade in analyze mode. [User must review the "analyze" results and fix them before running the "deploy" mode. 6. User should execute the playbook with deploy tag for autoupgrade to run in deploy mode.

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_dba/playbooks/vars/upgrade/upgrade_si_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_dba/playbooks/vars/upgrade/upgrade_si_vars.yml].
5. The freespace in the installer path of 19c homes should be more than 80GB. This is applicable when you want to apply Release Update patches (RU) along with Grid & Oracle Homes installation.
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_dba/playbooks/vars/upgrade/upgrade_si_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_dba/playbooks/vars/upgrade/upgrade_si_vars.yml file. The listed databases will be shut down during Single 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. Upgrade of Grid Infrastructure for RAC & RAC databases are NOT SUPPORTED as of this release.
11. Always run the playbooks in a vnc viewer to avoid ssh timeouts.
12. Following tags are provided:
    a. si_has_upgrade: This will invoke the role “si_has_upgrade”. Upgrades SI Grid Infrastructure to 19c from 12c.
    b. oracle_install: This will invoke the role “oracle_install”. Installs 19c Oracle Homes for database upgrades from 12c to 19c.
    c. predbupgrade: This will invoke prerequisite part of “db_upgrade” role which will do prechecks on the existing databases running on the lpar.
    d. analyze: This will invoke "autoupgrade" analyze mode which is a section of “db_upgrade” role.
    e. deploy: This will invoke a section of “db_upgrade” role which will invoke "autoupgrade" deploy mode, which is the core database upgrade mode.
13. These playbooks will create three directories (podba*) in /tmp. They should NOT be removed until the upgrade process completes otherwise it will compromise idempotency.
14. Try this on a non-production environment first before using it on a Production environment.

Upgrade scenarios:

1. **Full Stack Upgrade**: Upgrade Single Instance GI & all the databases running on ASM. Playbook must be executed twice as shown below. The first command will perform 19c Single 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_si.yml -i inventory.yml --ask-vault-pass --tags si_upgrade,oracle_install,predbupgrade,analyze
   2) ansible-playbook upgrade_si.yml -i inventory.yml --ask-vault-pass --tags deploy
   ```

2. **Other scenarios:**

   a) To upgrade only Single Instance Grid, run the following command.

   ```
   $ ansible-playbook upgrade_si.yml -i inventory.yml --ask-vault-pass --tags si_upgrade
   ```

   b) If the databases are running on JFS then Single Instance Grid upgrade is not required. In such case, skip the role “si_has_upgrade”

   ```
   $ ansible-playbook upgrade_si.yml -i inventory.yml --ask-vault-pass –skip-tags si_has_upgrade
   ```

   c) If 19c Oracle Homes are already installed, skip the role “oracle_install” and directly jump to “db_upgrade”.

   ```
   $ ansible-playbook upgrade_si.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_si.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_si.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_si.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_si.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 Single Instance GI to Oracle 19.21 along with one database. The s/w binaries and patches are placed in the Ansible controller. The Ansible playbook will extract it onto the target host before doing the upgrade.

Playbook upgrade_si.yml:

```
- hosts: ansible_db
  remote_user: root
  gather_facts: False
  vars_files:
      - vars/upgrade/upgrade_si_vars.yml
      - vars/vault.yml
  roles:
     - role: podba_si_has_upgrade
       tags: si_has_upgrade
     - role: podba_oracle_install
       tags: oracle_install
     - role: podba_db_upgrade
```

Update the variables file:

```
$ cat vars/upgrade/upgrade_si_vars.yml
# Section A - Update the Common Variables.
ora_binary_location: remote
ora_nfs_host: 129.40.76.1
ora_nfs_device: /repos
ora_nfs_filesystem: /repos

grid_sw: /home/ansible/binaries/V982588-01_193000_grid.zip
db_oracle_sw: /home/ansible/binaries/V982583-01_193000_db.zip
apply_ru: True
ru_stage: /backup/stage/RU
opatch_sw: /home/ansible/binaries/opatch-12.2.0.1.42_p6880880_190000_AIX64-5L.zip
ru_zip: /home/ansible/binaries/p35642822_190000_AIX64-5L_RU19.21.zip
autoupgrade_util_remote: /home/ansible/binaries/autoupgrade.jar
autoupgrade_util: /home/oracle/autoupgrade.jar

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

# Section B - Update the Single Instance Grid Variables
grid_home_prev: /u01/base/grid12c
grid_home: /u02/base/grid19c
gi_oracle_base: /u02/base
grid_user: oracle
grid_group: oinstall
gi_osdba_group: dba
gi_osasm_group: oinstall
gi_osoper_group:
gi_cluster_name:

# Section C - Update the Variables for Oracle 19c RDBMS installation
db_oracle_base: /u02/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: /u02/autoupgrade_1
databases:
    - source_db_name: podba
      source_db_home: /u01/base/db12c
      target_db_home: /u02/base/db19c
      log_dir: /u02/autoupgrade_1/dbupgrdlogs
      start_time: NOW
      restoration: 'yes'
      upgrade_node: localhost
      run_utlrp: '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 Single 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
 ansible_db
```

```
[ansible@localhost playbooks]$ ansible-playbook upgrade_si.yml -i inventory.yml --ask-vault-pass --tags si_upgrade,oracle_install,predbupgrade,analyze

PLAY [ansible_db]
**************************************************************

TASK [ibm.power_aix_oracle_dba.podba_si_has_upgrade : Checking if Upgrade
Task was already done] ***
ok: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_si_has_upgrade : Fail if Upgrade Task
was already done] ***
skipping: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_si_has_upgrade : Checking if 19c GI is
installed] ***
ok: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_si_has_upgrade : Creating Temp
Directory] ***
ok: [ansible_db] => (item=logs)
ok: [ansible_db] => (item=done)
ok: [ansible_db] => (item=scripts)

TASK [ibm.power_aix_oracle_dba.podba_si_has_upgrade : Checking if prechecks
script is already run] ***
ok: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_si_has_upgrade : Checking current HAS
Version] ***
ok: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_si_has_upgrade : Setting fact - HAS
Version] ***
ok: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_si_has_upgrade : Templating out
prechecks.sh script] ***
changed: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_si_has_upgrade : Executing
prechecks.sh script] ***
changed: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_si_has_upgrade : Prechecks Script
Output] ***
ok: [ansible_db] => {
    "precheck_out.stdout_lines": [
        "Prechecks completed successfully."
    ]
}

TASK [ibm.power_aix_oracle_dba.podba_si_has_upgrade : Checking if GI S/W is
extracted] ***
ok: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_si_has_upgrade : Checking if RU is
extracted] ***
ok: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_si_has_upgrade : Checking if Opatch is
extracted] ***
ok: [ansible_db]
```

```
TASK [ibm.power_aix_oracle_dba.podba_si_has_upgrade : Creating NFS
filesystem from nfshost.] ***
skipping: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_si_has_upgrade : Creating GI home
directory] ***
ok: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_si_has_upgrade : Creating RU Stage
Directory] ***
ok: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_si_has_upgrade : Extracting GI S/W
(Remote)] ***
changed: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_si_has_upgrade : Creating the done
file] ***
changed: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_si_has_upgrade : Creating the done
file] ***
skipping: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_si_has_upgrade : Extracting RU S/W
(Remote)] ***
ok: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_si_has_upgrade : Creating the done
file] ***
changed: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_si_has_upgrade : Creating the done
file] ***
skipping: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_si_has_upgrade : Backup Opatch]
***********
changed: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_si_has_upgrade : Extracting OPatch
(Remote)] ***
changed: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_si_has_upgrade : Creating the done
file] ***
changed: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_si_has_upgrade : Backup Opatch]
***********
skipping: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_si_has_upgrade : Extracting OPatch
(Local|NFS)] ***
skipping: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_si_has_upgrade : Creating the done
file] ***
skipping: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_si_has_upgrade : Checking freespace in
GI Installation Path /u02/base/grid19c] ***
ok: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_si_has_upgrade : fail]
********************
skipping: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_si_has_upgrade : Checking if cluvfy was
already executed] ***
ok: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_si_has_upgrade : Running Cluvfy]
**********
changed: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_si_has_upgrade : debug]
*******************
skipping: [ansible_db]
```

```
TASK [ibm.power_aix_oracle_dba.podba_si_has_upgrade : Executing rootpre.sh]
****
changed: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_si_has_upgrade : Templating out GI
Response File] ***
changed: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_si_has_upgrade : Templating out
grid_install.sh Script] ***
changed: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_si_has_upgrade : Setting Up New 19c
GI for HAS] ***
changed: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_si_has_upgrade : debug]
*******************
ok: [ansible_db] => {
    "new_grid_out.stdout": "gridSetup.sh completed successfully.\nRU Patch
35642822 applied successfully"
}

TASK [ibm.power_aix_oracle_dba.podba_si_has_upgrade : Checking if
rootupgrade.sh script was already executed] ***
ok: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_si_has_upgrade : Stopping database
services before running rootupgrade.sh] ***
changed: [ansible_db] => (item=None)
changed: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_si_has_upgrade : Templating out
grid_upgrade.sh (rootupgrade.sh) script] ***
changed: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_si_has_upgrade : Executing
grid_upgrade.sh script] ***
changed: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_si_has_upgrade : ansible.builtin.debug]
***
ok: [ansible_db] => {
    "grid_upgrade_out.stdout": "Rootupgrade.sh completed successfully"
}

TASK [ibm.power_aix_oracle_dba.podba_si_has_upgrade : Post Grid Upgrade
Steps | Copying config_tools.sh] ***
changed: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_si_has_upgrade : Post GI Upgrade
Steps| Executing the Config Tools script] ***
changed: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_si_has_upgrade : ansible.builtin.debug]
***
ok: [ansible_db] => {
    "config_tools_out.stdout": "gridSetup.sh -executeConfigTools completed
successfully"
}

TASK [ibm.power_aix_oracle_dba.podba_si_has_upgrade : Post GI Upgrade
Steps | Checking if Config Tools is successful.] ***
ok: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_si_has_upgrade : Post GI Upgrade
Steps| Status of GI Services] ***
changed: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_si_has_upgrade : ansible.builtin.debug]
***
ok: [ansible_db] => {
    "check_services.stdout_lines": [
        "NAME=ora.cssd",
        "TYPE=ora.cssd.type",
        "TARGET=ONLINE",
        "STATE=ONLINE on ansible_db"
    ]
}
```

```
TASK [ibm.power_aix_oracle_dba.podba_si_has_upgrade : Post GI Upgrade
Steps | Starting the database services] ***
changed: [ansible_db] => (item=None)
changed: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_oracle_install : Creating Temp Directory
| /tmp/podba_install_temp] ***
ok: [ansible_db] => (item=done)
ok: [ansible_db] => (item=scripts)
ok: [ansible_db] => (item=logs)

TASK [ibm.power_aix_oracle_dba.podba_oracle_install : Reading Oracle
Inventory File] ***
ok: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_oracle_install : Setting Fact for
Inventory File] ***
ok: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_oracle_install : Preparing Oracle Homes
List for Installation] ***
ok: [ansible_db] => (item=None)
ok: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_oracle_install : Checking if DB S/W is
extracted] ***
ok: [ansible_db] => (item={'oh': 'u02basedb19c', 'oracle_home':
'/u02/base/db19c'})

TASK [ibm.power_aix_oracle_dba.podba_oracle_install : Checking if DB OPatch is
extracted] ***
ok: [ansible_db] => (item={'oh': 'u02basedb19c', 'oracle_home':
'/u02/base/db19c'})

TASK [ibm.power_aix_oracle_dba.podba_oracle_install : Creating DB Home
Directory] ***
changed: [ansible_db] => (item={'oh': 'u02basedb19c', 'oracle_home':
'/u02/base/db19c'})

TASK [ibm.power_aix_oracle_dba.podba_oracle_install : Extracting DB S/W
(Remote)] ***
changed: [ansible_db] => (item={'oh': 'u02basedb19c', 'oracle_home':
'/u02/base/db19c'})

TASK [ibm.power_aix_oracle_dba.podba_oracle_install : Creating the done file]
***
changed: [ansible_db] => (item={'oh': 'u02basedb19c', 'oracle_home':
'/u02/base/db19c'})

TASK [ibm.power_aix_oracle_dba.podba_oracle_install : Checking if RU is
extracted] ***
ok: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_oracle_install : Checking if Opatch is
extracted] ***
ok: [ansible_db] => (item={'oh': 'u02basedb19c', 'oracle_home':
'/u02/base/db19c'})

TASK [ibm.power_aix_oracle_dba.podba_oracle_install : Creating RU Stage
Directory] ***
ok: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_oracle_install : Extracting RU S/W
(Remote)] ***
ok: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_oracle_install : Creating the done file]
***
changed: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_oracle_install : Extracting DB RU patch
(Local|NFS)] ***
skipping: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_oracle_install : Creating the done file]
***
skipping: [ansible_db]
```

```
TASK [ibm.power_aix_oracle_dba.podba_oracle_install : Backup Opatch]
***********
changed: [ansible_db] => (item=/u02/base/db19c)

TASK [ibm.power_aix_oracle_dba.podba_oracle_install : Extracting Opatch
(Remote)] ***
changed: [ansible_db] => (item=/u02/base/db19c)

TASK [ibm.power_aix_oracle_dba.podba_oracle_install : Extracting Opatch
(Local|NFS)] ***
skipping: [ansible_db] => (item=/u02/base/db19c)
skipping: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_oracle_install : Creating done file]
******
changed: [ansible_db] => (item=/u02/base/db19c)

TASK [ibm.power_aix_oracle_dba.podba_oracle_install : Copying Oracle RDBMS
Install response file] ***
changed: [ansible_db] => (item={'oh': 'u02basedb19c', 'oracle_home':
'/u02/base/db19c'})

TASK [ibm.power_aix_oracle_dba.podba_oracle_install : Copying
oracle_install.sh] ***
changed: [ansible_db] => (item={'oh': 'u02basedb19c', 'oracle_home':
'/u02/base/db19c'})

TASK [ibm.power_aix_oracle_dba.podba_oracle_install : Installing 19c RDBMS.]
***
changed: [ansible_db] => (item={'oh': 'u02basedb19c', 'oracle_home':
'/u02/base/db19c'})

TASK [ibm.power_aix_oracle_dba.podba_oracle_install : Oracle Install Output]
***
ok: [ansible_db] => (item=['Oracle install done', 'RU Patch 35642822 applied
successfully']) => {
    "msg": [
        "Oracle install done",
        "RU Patch 35642822 applied successfully"
    ]
}

TASK [ibm.power_aix_oracle_dba.podba_oracle_install : Executing root.sh]
*******
changed: [ansible_db] => (item={'oh': 'u02basedb19c', 'oracle_home':
'/u02/base/db19c'})

TASK [ibm.power_aix_oracle_dba.podba_db_upgrade : Stage Autoupgrade
Utility] ***
changed: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_db_upgrade : Creating Temp Directory
| /tmp/podba_db_upgrade] ***
ok: [ansible_db] => (item=None)
ok: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_db_upgrade : Autoupgrade Full DB |
Checking source database name in /etc/oratab file] ***
changed: [ansible_db] => (item=None)
changed: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_db_upgrade : Fail if source database
name doesn't exist in oratab file.] ***
skipping: [ansible_db] => (item=None)
skipping: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_db_upgrade : Autoupgrade Full DB |
Checking the status of the Source Database] ***
changed: [ansible_db] => (item=None)
changed: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_db_upgrade : Autoupgrade Full DB |
Fail if Source database is not running.] ***
skipping: [ansible_db] => (item=None)
skipping: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_db_upgrade : Autoupgrade Non-CDB to
PDB | Checking target database name in /etc/oratab file] ***
changed: [ansible_db] => (item=None)
changed: [ansible_db]
```

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

TASK [ibm.power_aix_oracle_dba.podba_db_upgrade : Autoupgrade Full DB |
Generating response file for autoupgrade] ***
changed: [ansible_db] => (item=None)
changed: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_db_upgrade : Autoupgrade - Analyze]
*******
changed: [ansible_db] => (item={'source_db_name': 'podba', 'source_db_home':
'/u01/base/db12c', 'target_db_home': '/u02/base/db19c', 'log_dir':
'/u02/autoupgrade_1/dbupgrdlogs', 'start_time': 'NOW', 'restoration': 'yes',
'upgrade_node': 'localhost', 'run_utlrp': 'yes', 'timezone_upg': 'yes',
'target_cdb_name': None, 'target_pdb_name': None})

TASK [ibm.power_aix_oracle_dba.podba_db_upgrade : debug]
***********************
ok: [ansible_db] => {
    "analyze.results[0].stdout_lines": [
        "AutoUpgrade 24.4.240426 launched with default internal options",
        "Processing config file ...",
        "+--------------------------------+",
        "| Starting AutoUpgrade execution |",
        "+--------------------------------+",
        "1 CDB(s) plus 2 PDB(s) will be analyzed",
        "Job 100 database podba",
        "+----+-------+---------+---------+-------+----------+-------+----------------------------+",
        "|Job#|DB_NAME|    STAGE|OPERATION| STATUS|START_TIME|UPDATED|
MESSAGE|",
        "+----+-------+---------+---------+-------+----------+-------+----------------------------+",
        "| 100|  podba|PRECHECKS|EXECUTING|RUNNING|  04:43:21| 0s
ago|Loading database information|",
        "+----+-------+---------+---------+-------+----------+-------+----------------------------+",
        "Total jobs 1",
        "",
        "Job 100 completed",
        "------------------- Final Summary --------------------",
        "Number of databases            [ 1 ]",
        "",
        "Jobs finished                  [1]",
        "Jobs failed                    [0]",
        "",
        "Please check the summary report at:",
        "/u02/autoupgrade_1/cfgtoollogs/upgrade/auto/status/status.html",
        "/u02/autoupgrade_1/cfgtoollogs/upgrade/auto/status/status.log"
    ]
}
```

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 ansible_db
```

At this point, the following are completed:

- GI upgrade.
- Oracle 19c RDBMS installation.
- Autoupgrade analyze mode.

The next part is the Autoupgrade deploy mode to upgrade the DB.

Sample output – DB upgrade continuation:

```
[ansible@localhost playbooks]$ ansible-playbook upgrade_si.yml -i inventory.yml --ask-vault-pass --tags
deploy Vault password:

PLAY [ansible_db] ***************************************************************

TASK [ibm.power_aix_oracle_dba.podba_db_upgrade : Autoupgrade - Deploy] ********
changed: [ansible_db] => (item={'source_db_name': 'podba', 'source_db_home': '/u01/base/db12c', 'target_db_home': '/u02/base/db19c', 'log_dir':
'/u02/autoupgrade_1/dbupgrdlogs', 'start_time': 'NOW', 'restoration': 'yes', 'upgrade_node': 'localhost', 'run_utlrp': 'yes', 'timezone_upg': 'yes', 'target_cdb_name':
None, 'target_pdb_name': None})

TASK [ibm.power_aix_oracle_dba.podba_db_upgrade : debug] ***********************
ok: [ansible_db] => {
    "deploy.results[0].stdout_lines": [
        "AutoUpgrade 24.4.240426 launched with default internal options",
        "Processing config file ...",
        "+--------------------------------+",
        "| Starting AutoUpgrade execution |",
        "+--------------------------------+",
        "1 CDB(s) plus 2 PDB(s) will be processed",
        "Job 101 database podba",
        "+----+-------+---------+---------+-------+----------+-------+----------------------------+",
        "|Job#|DB_NAME|    STAGE|OPERATION| STATUS|START_TIME|UPDATED|                     MESSAGE|",
        "+----+-------+---------+---------+-------+----------+-------+----------------------------+",
        "| 101|  podba|PRECHECKS|EXECUTING|RUNNING|  04:44:13| 0s ago|Loading database information|",
        "+----+-------+---------+---------+-------+----------+-------+----------------------------+",
        "Total jobs 1",
        "",
        "+----+-------+---------+---------+-------+----------+-------+------------------+",
        "|Job#|DB_NAME|    STAGE|OPERATION| STATUS|START_TIME|UPDATED|           MESSAGE|",
        "+----+-------+---------+---------+-------+----------+-------+------------------+",
        "| 101|  podba|PREFIXUPS|EXECUTING|RUNNING|  04:44:13|18s ago|Refreshing DB info|",
        "+----+-------+---------+---------+-------+----------+-------+------------------+",
        "Total jobs 1",
        "",
        "+----+-------+-----+---------+-------+----------+-------+------------------------+",
        "|Job#|DB_NAME|STAGE|OPERATION| STATUS|START_TIME|UPDATED|                 MESSAGE|",
        "+----+-------+-----+---------+-------+----------+-------+------------------------+",
        "| 101|  podba|DRAIN|EXECUTING|RUNNING|  04:44:13|18s ago|Disabling RAC if present|",
        "+----+-------+-----+---------+-------+----------+-------+------------------------+",
        "Total jobs 1",
        "",
        "+----+-------+-----+---------+-------+----------+-------+----------------------+",
        "|Job#|DB_NAME|STAGE|OPERATION| STATUS|START_TIME|UPDATED|               MESSAGE|",
        "+----+-------+-----+---------+-------+----------+-------+----------------------+",
        "| 101|  podba|DRAIN|EXECUTING|RUNNING|  04:44:13| 3s ago|Shutting down database|",
        "+----+-------+-----+---------+-------+----------+-------+----------------------+",
        "Total jobs 1",
        "",
        "+----+-------+---------+---------+-------+----------+-------+-------------------+",
        "|Job#|DB_NAME|    STAGE|OPERATION| STATUS|START_TIME|UPDATED|            MESSAGE|",
        "+----+-------+---------+---------+-------+----------+-------+-------------------+",
        "| 101|  podba|DBUPGRADE|EXECUTING|RUNNING|  04:44:13| 4s ago|0%Upgraded CDB$ROOT|",
        "+----+-------+---------+---------+-------+----------+-------+-------------------+",
        "Total jobs 1",
        "+----+-------+---------+---------+-------+----------+-------+--------------------+",
        "|Job#|DB_NAME|    STAGE|OPERATION| STATUS|START_TIME|UPDATED|             MESSAGE|",
        "+----+-------+---------+---------+-------+----------+-------+--------------------+",
        "| 101|  podba|DBUPGRADE|EXECUTING|RUNNING|  04:44:13|68s ago|91%Upgraded CDB$ROOT|",
        "+----+-------+---------+---------+-------+----------+-------+--------------------+",
        "Total jobs 1",
        "+----+-------+---------+---------+-------+----------+-------+-------------------+",
        "|Job#|DB_NAME|    STAGE|OPERATION| STATUS|START_TIME|UPDATED|            MESSAGE|",
        "+----+-------+---------+---------+-------+----------+-------+-------------------+",
        "| 101|  podba|DBUPGRADE|EXECUTING|RUNNING|  04:44:13| 0s ago|0%Upgraded PODBAPDB|",
        "+----+-------+---------+---------+-------+----------+-------+-------------------+",
        "|Job#|DB_NAME|    STAGE|OPERATION| STATUS|START_TIME|UPDATED|             MESSAGE|",
        "+----+-------+---------+---------+-------+----------+-------+--------------------+",
        "| 101|  podba|DBUPGRADE|EXECUTING|RUNNING|  04:44:13|27s ago|97%Upgraded PODBAPDB|",
        "+----+-------+---------+---------+-------+----------+-------+--------------------+",
        "Total jobs 1",
        "",
        "+----+-------+---------+---------+-------+----------+-------+--------------------+",
        "|Job#|DB_NAME|    STAGE|OPERATION| STATUS|START_TIME|UPDATED|             MESSAGE|",
        "+----+-------+---------+---------+-------+----------+-------+--------------------+",
        "| 101|  podba|DBUPGRADE|EXECUTING|RUNNING|  04:44:13|73s ago|95%Upgraded PDB$SEED|",
        "+----+-------+---------+---------+-------+----------+-------+--------------------+",
        "Total jobs 1",
        "+----+-------+--------+---------+-------+----------+-------+-------------------+",
        "|Job#|DB_NAME|   STAGE|OPERATION| STATUS|START_TIME|UPDATED|            MESSAGE|",
        "+----+-------+--------+---------+-------+----------+-------+-------------------+",
        "| 101|  podba|DISPATCH|EXECUTING|RUNNING|  04:44:13|20s ago|Restarting Database|",
        "+----+-------+--------+---------+-------+----------+-------+-------------------+",
        "Total jobs 1",
        "+----+-------+----------+---------+-------+----------+-------+------------------+",
        "|Job#|DB_NAME|     STAGE|OPERATION| STATUS|START_TIME|UPDATED|           MESSAGE|",
        "+----+-------+----------+---------+-------+----------+-------+------------------+",
        "| 101|  podba|POSTFIXUPS|EXECUTING|RUNNING|  04:44:13| 0s ago|Refreshing DB info|",
        "+----+-------+----------+---------+-------+----------+-------+------------------+",
        "Total jobs 1",
        "",
        "+----+-------+----------+---------+-------+----------+-------+----------------------------------------+",
        "|Job#|DB_NAME|     STAGE|OPERATION| STATUS|START_TIME|UPDATED|                                 MESSAGE|",
        "+----+-------+----------+---------+-------+----------+-------+----------------------------------------+",
        "| 101|  podba|SYSUPDATES|EXECUTING|RUNNING|  04:44:13|12s ago|RAC configurations have finished success|",
        "+----+-------+----------+---------+-------+----------+-------+----------------------------------------+",
        "Total jobs 1",
        "",
        "+----+-------+----------+---------+-------+----------+-------+----------------------------------------+",
        "|Job#|DB_NAME|     STAGE|OPERATION| STATUS|START_TIME|UPDATED|                                 MESSAGE|",
        "+----+-------+----------+---------+-------+----------+-------+----------------------------------------+",
        "| 101|  podba|SYSUPDATES|EXECUTING|RUNNING|  04:44:13| 1s ago|RAC configurations have finished success|",
        "+----+-------+----------+---------+-------+----------+-------+----------------------------------------+",
        "Total jobs 1",
        "",
        "+----+-------+----------+---------+-------+----------+-------+----------------------------------------+",
        "|Job#|DB_NAME|     STAGE|OPERATION| STATUS|START_TIME|UPDATED|                                 MESSAGE|",
        "+----+-------+----------+---------+-------+----------+-------+----------------------------------------+",
        "| 101|  podba|SYSUPDATES|EXECUTING|RUNNING|  04:44:13| 2s ago|RAC configurations have finished success|",
        "+----+-------+----------+---------+-------+----------+-------+----------------------------------------+",
        "Total jobs 1",
        "",
        "Job 101 completed",
        "------------------- Final Summary --------------------",
        "Number of databases            [ 1 ]",
        "",
        "Jobs finished                  [1]",
        "Jobs failed                    [0]",
        "Jobs restored                  [0]",
        "Jobs pending                   [0]",
        "",
        "---- Drop GRP at your convenience once you consider it is no longer needed ----",
        "Drop GRP from podba: drop restore point AUTOUPGRADE_9212_PODBA121020",
        "",
        "",
        "Please check the summary report at:",
        "/u02/autoupgrade_1/cfgtoollogs/upgrade/auto/status/status.html",
        "/u02/autoupgrade_1/cfgtoollogs/upgrade/auto/status/status.log"
    ]
}

TASK [ibm.power_aix_oracle_dba.podba_db_upgrade : Checking the status of the DBs] ***
changed: [ansible_db] => (item=None)
changed: [ansible_db]

TASK [ibm.power_aix_oracle_dba.podba_db_upgrade : Status of the DBs] ***********
ok: [ansible_db] => (item=podba) => {
    "msg": " Status: podba Database is running."
}

PLAY RECAP *********************************************************************
ansible_db                 : ok=13   changed=8    unreachable=0    failed=0    skipped=4    rescued=0    ignored=0
```

To execute this playbook from AAP2 use the playbook upgade_si_aap2.yml. An example document is provided here:
https://github.com/IBM/ansible-power-aix-oracle-dba/blob/main/docs/PowerODBA_using_AAP2.pdf