New release PODBA 2.0

Introducing three new roles to upgrade Single Instance Grid Infrastructure & the databases.

Scenario: Upgrade 12c Single Instance Grid Infrastructure and databases to 19c version.

System configuration: The following configuration has been used to test this scenario.

1. Ansible controller - Linux on x86

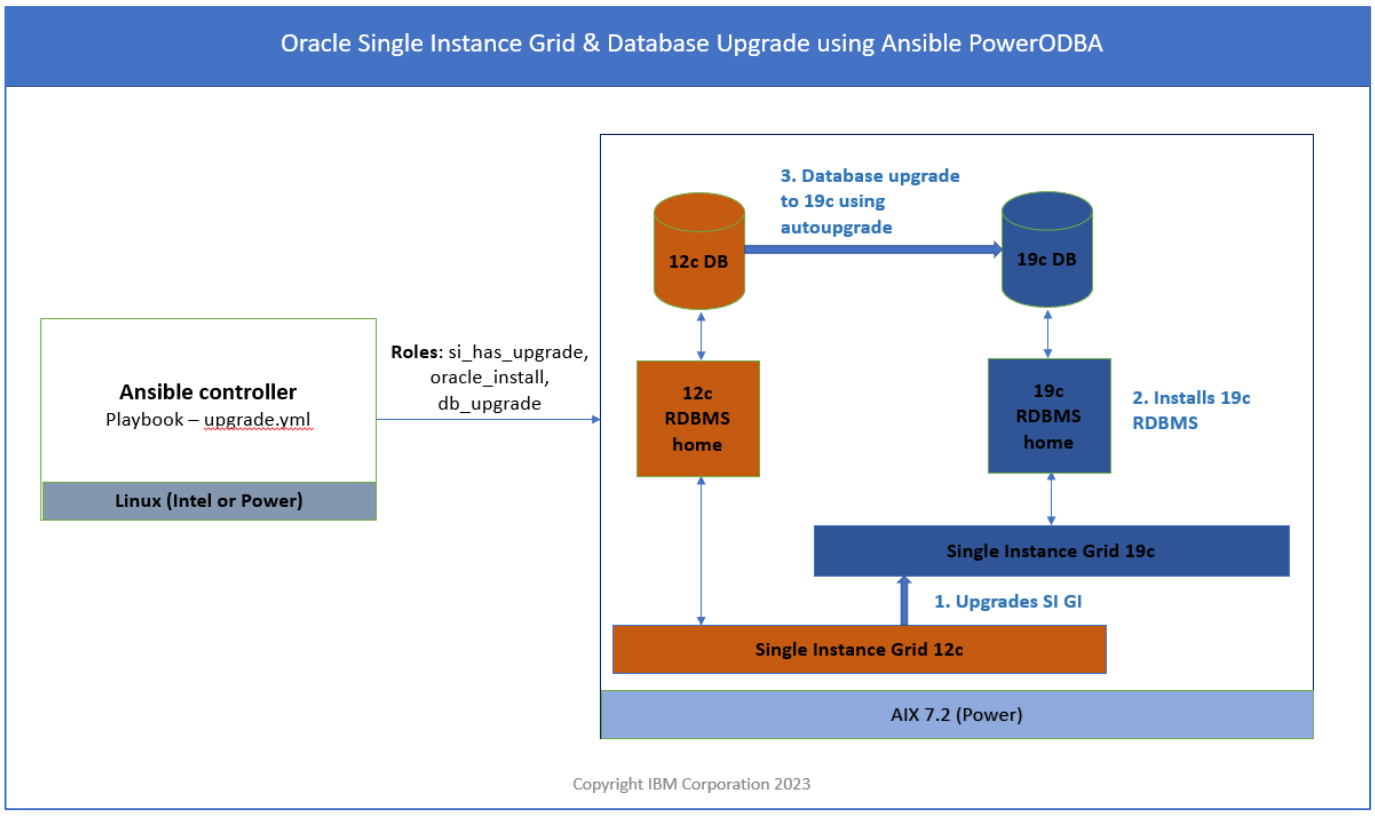
Operating System: Redhat linux 8.5
Ansible engine: 2.12.2

2. Target lpar - AIX server

Operating system: AIX 72 TL 5 SP 4
Oracle Grid & Database: 12.1.2.0 (APR 2022 PSU)
CPU: 4
Memory: 64GB

Prerequisite steps:

1. Stage the 19c RDBMS home software, Release Update patch & latest Opatch utility zip files in a directory on the target lpar, this path must be specified to the variable “sw_stage” in the variables file [ansible-power-aix-oracle-dba/playbooks/vars/upgrade/upgrade_si_vars.yml].
2. Stage latest autoupgrade.jar file [AutoUpgrade Tool (Doc ID 2485457.1)] in a directory on the target lpar, this path must be specified to the variable “autoupgrade_file_loc” in the variables file [ansible-power-aix-oracle-dba/playbooks/vars/upgrade/upgrade_si_vars.yml].



Playbook:

```
- hosts: ansible_db                  # Provide the name of the target lpar registered in ansible_inventory.
  remote_user: root                  # This needs to be run by "root" user.
  gather_facts: False
  vars_files:
     - /home/ansible/shiva/vars/globalvars.yml
     - /home/ansible/shiva/vars/vault.yml
  roles:
    - role: si_has_upgrade
      tags: si_has_upgrade
    - role: oracle_install
      tags: oracle_install
    - role: db_upgrade
```

Roles used by the playbook:

- si_has_upgrade: Upgrades Single Instance 12c Grid to 19c.
- oracle_install: Installs 19c Oracle database home with release update patch
- db_upgrade: Upgrades the database using autoupgrade.jar utility. Provides following upgrade features.
    - Full Database upgrade: Existing 12c Non-CDB (or) CDB database upgrade to 19c.
    - Non-container to Pluggable Database upgrade: Existing 12c Non-CDB will be plugged and upgraded into 19c Container Database. It won’t create new databases in 19c environment, users have to create a new container database in 19c and use this feature.

Role: si_has_upgrade

- Checks for existence of new 19c grid installations.
- Checks for source (12c) grid setup.
- Checks the required value of maxuproc value.
- Checks for mandatory patch on source (12c) grid home for 19c upgrade.
- Extracts the 19c Grid software.
- Backups up OPatch and extracts latest OPatch when apply_ru is true.
- Extracts Release update patch when apply_ru is true
- Checks freespace in Grid home path and fails if freespace is less than 80GB.
- Runs cluvfy.sh and pauses in case missing/failed items are found.
- Executes rootpre.sh
- Creates & copies grid_upgrade.rsp response file from template.
- Executes gridSetup.sh (with upgrade option) in silent mode along with -applyRU when apply_ru is true or executes without applyRU option, lists the log files upon completion.
- Stops the databases from 12c DB home provided in the variables file.
- Executes rootupgrade.sh and lists the log files.
- Executes gridSetup.sh -silent -executeConfigTools and lists the log files upon completion.
- Checks the status of grid services.
- Starts the databases from 12c DB home which were stopped before upgrade.

Role: oracle_install

- Checks for any failed 19c RDBMS installations.
- Checks the required value of maxuproc value.
- Extracts the 19c RDBMS software.
- Backups up OPatch and extracts latest OPatch when apply_ru is true.
- Extracts Release update patch when apply_ru is true
- Checks freespace in Grid home path and fails if freespace is less than 80GB.
- Executes rootpre.sh
- Creates & copies oracle_install.rsp response file from template.
- Installs 19c RDBMS.
- Executes root.sh

Role: db_upgrade

- Creates a stage directory to place the autoupgrade configuration file.
- Checks DB name in oratab.
- Checks DB is up and running.
- Generates autoupgrade configuration file from template.
- Executes autoupgrade in analyze mode. [User must review the “analyze” results and fix them.
- User will execute the playbook with deploy tag for autoupgrade to run in deploy mode.

Steps to execute

1. Open the variables file **upgrade_si_vars.yml** under - **ansible-power-aix-oracle-dba/playbooks/vars/upgrade** and update all the variables as per your environment. Instructions to update the variables are provided in the variables file itself.
2. Open **vault.yml** file under - **ansible-power-aix-oracle-dba/playbooks/vars** and update the asm sys password. Encrypt the file using ansible-encrypt. 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)
3. Open the playbook file **upgrade_si.yml** under - **ansible-power-aix-oracle-dba/playbooks** and update your target lpar hostname as mentioned in your ansible inventory file.
4. For the database upgrade – we are using the following tags to segregate the execution between “Analyze” & “Deploy” modes.
    a. predbupgrade - this will perform prechecks.
    b. analyze - this will execute autoupgrade in analyze mode.
    c. deploy - this will execute autoupgrade in deploy mode.
5. The playbook should be executed twice as shown below. 1st time execution 5 (a) will run SIHA upgrade, Oracle Install & Database upgrade in analyze mode. Users should review the logs and act on the errors/recommendations reported by the autoupgrade tool and rerun the playbook 5 (b)
    a. ansible-playbook upgrade_si.yml -i inventory.yml --ask-vault-pass --tags si_upgrade,ora_install,predbupgrade,analyze
    b. ansible-playbook upgrade_si.yml -i inventory.yml --ask-vault-pass --tags deploy
6. To execute individual roles, kindly use tags. Each role has a tag with the same name. For example: To execute only SI Grid Upgrade, run the following command.
    a. ansible-playbook upgrade_si.yml -i inventory.yml --ask-vault-pass --tags si_upgrade

Output from Step 5(a):

SIHA Version before performing 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@x134vm236 playbooks]$ ansible-playbook upgrade_si.yml -i inventory.yml --ask-vault-pass --tags
si_has_upgrade,oracle_install,predbupgrade,analyze
Vault password:
[WARNING]: running playbook inside collection ibm.power_aix_oracle_dba

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

TASK [si_has_upgrade : Checking if Upgrade Task was already done] ********************************************
[WARNING]: Platform aix on host ansible_db is using the discovered Python interpreter at /usr/bin/python2.7, but future installation
of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-
core/2.12/reference_appendices/interpreter_discovery.html for more information.
ok: [ansible_db]

TASK [si_has_upgrade : fail]
**********************************************************************************************************
skipping: [ansible_db]

TASK [si_has_upgrade : Checking if 19c Grid Infra is already setup] ******************************************
ok: [ansible_db]

TASK [si_has_upgrade : Checking for failed installations] ********************************************
ok: [ansible_db]

TASK [si_has_upgrade : Cleaning up failed installations | Grid Home] ********************************************
skipping: [ansible_db]

TASK [si_has_upgrade : Cleaning up failed installations | podba_temp] ********************************************
skipping: [ansible_db]

TASK [si_has_upgrade : Creating Temp Directory]
**********************************************************************************
ok: [ansible_db]

TASK [si_has_upgrade : Checking if prechecks already run] ********************************************
ok: [ansible_db]

TASK [si_has_upgrade : Check HAS Version]
**********************************************************************************************
ok: [ansible_db]

TASK [si_has_upgrade : Getting HAS Version]
**********************************************************************************************
ok: [ansible_db]

TASK [si_has_upgrade : Copying prechecks.sh]
**********************************************************************************************
changed: [ansible_db]

TASK [si_has_upgrade : Executing prechecks.sh]
**********************************************************************************************
changed: [ansible_db]

TASK [si_has_upgrade : debug]
**********************************************************************************************************
ok: [ansible_db] => {
    "precheck_out.stdout_lines": [
        "Prechecks completed successfully."
    ]
}

TASK [si_has_upgrade : Copying sw_extract.sh]
**********************************************************************************************
changed: [ansible_db]

TASK [si_has_upgrade : Execute sw_extract.sh]
**********************************************************************************************
changed: [ansible_db]

TASK [si_has_upgrade : Checking freespace in Grid Install Path] ********************************************
```

```
ok: [ansible_db]

TASK [si_has_upgrade : fail]
*********************************************************************************************************
skipping: [ansible_db]

TASK [si_has_upgrade : Checking if cluvfy already executed] ********************************************************************
ok: [ansible_db]

TASK [si_has_upgrade : Running Cluvfy]
*********************************************************************************************
changed: [ansible_db]

TASK [si_has_upgrade : debug]
*********************************************************************************************
skipping: [ansible_db]

TASK [si_has_upgrade : ansible.builtin.pause] ***********************************************************************************
skipping: [ansible_db]

TASK [si_has_upgrade : Executing rootpre.sh]
*****************************************************************************
changed: [ansible_db]

TASK [si_has_upgrade : Copying grid response file] ******************************************************************************
changed: [ansible_db]

TASK [si_has_upgrade : Copying grid_install.sh] *********************************************************************************
changed: [ansible_db]

TASK [si_has_upgrade : Setting Up new 19c Grid for HAS] *************************************************************************
changed: [ansible_db]

TASK [si_has_upgrade : debug]
*********************************************************************************************
ok: [ansible_db] => {
    "new_grid_out.stdout": "Grid install done"
}

TASK [si_has_upgrade : Checking if rootupgrade is already executed] *************************************************************
ok: [ansible_db]

TASK [si_has_upgrade : Stopping database services before Root Upgrade] **********************************************************
changed: [ansible_db] => (item=podba)

TASK [si_has_upgrade : Copying grid_upgrade.sh]
*****************************************************************************
changed: [ansible_db]

TASK [si_has_upgrade : Executing grid_upgrade.sh] *******************************************************************************
changed: [ansible_db]

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

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

TASK [si_has_upgrade : Post Grid Upgrade Steps| Executing Config Tools] *********************************************************
changed: [ansible_db]

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

TASK [si_has_upgrade : Post Grid Upgrade Steps| Checking the Grid Services] *****************************************************
changed: [ansible_db]

TASK [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 [si_has_upgrade : Post Grid Upgrade Steps | Starting database services] ***********************************************************
ok: [ansible_db]

TASK [si_has_upgrade : Starting database services] ***********************************************************************************
changed: [ansible_db] => (item=podba)

TASK [oracle_install : Checking if Oracle 19c is already installed] ******************************************************************
ok: [ansible_db]

TASK [oracle_install : fail] *****************************************************************************************************************
skipping: [ansible_db]

TASK [oracle_install : Checking for failed installations] ************************************************************************
ok: [ansible_db]

TASK [oracle_install : Cleaning up failed installations | Oracle DB Home] ***********************************************************
skipping: [ansible_db]

TASK [oracle_install : Cleaning up failed installations | podba_db_temp] ************************************************************
skipping: [ansible_db]

TASK [oracle_install : Creating Temp Directory | /tmp/podba_db_temp] *****************************************************************
ok: [ansible_db]

TASK [oracle_install : Checking if prechecks already run] ************************************************************************
ok: [ansible_db]

TASK [oracle_install : Copying prechecks.sh] *************************************************************************************
changed: [ansible_db]

TASK [oracle_install : Executing prechecks.sh] ***********************************************************************************
changed: [ansible_db]

TASK [oracle_install : debug]
***********************************************************************************************
ok: [ansible_db] => {
    "precheck_out.stdout_lines": [
        "Prechecks completed successfully."
    ]
}

TASK [oracle_install : Copying sw_extract.sh] ************************************************************************************
changed: [ansible_db]

TASK [oracle_install : Execute sw_extract.sh] ************************************************************************************
changed: [ansible_db]

TASK [oracle_install : Checking freespace in Grid Install Path] ******************************************************************
ok: [ansible_db]

TASK [oracle_install : fail] *****************************************************************************************************************
skipping: [ansible_db]

TASK [oracle_install : Executing rootpre.sh] *************************************************************************************
changed: [ansible_db]

TASK [oracle_install : Copying Oracle RDBMS Install response file] ***************************************************************
changed: [ansible_db]

TASK [oracle_install : Copying oracle_install.sh] ********************************************************************************
changed: [ansible_db]

TASK [oracle_install : Installing 19c RDBMS] *************************************************************************************
changed: [ansible_db]

TASK [oracle_install : debug]
***********************************************************************************************
ok: [ansible_db] => {
    "oracle_install.stdout": "Oracle install done"
}

TASK [oracle_install : Executing root.sh] ****************************************************************************************
changed: [ansible_db]
```

```
TASK [db_upgrade : Checking database name in /etc/oratab file] ********************************************************************
changed: [ansible_db]

TASK [db_upgrade : fail]
*****************************************************************************************************************
skipping: [ansible_db]

TASK [db_upgrade : Checking status of the Database] ********************************************************************************
changed: [ansible_db]

TASK [db_upgrade : fail]
*****************************************************************************************************************
skipping: [ansible_db]

TASK [db_upgrade : Creating Autoupgrade Stage directory] ***************************************************************************
changed: [ansible_db]

TASK [db_upgrade : DB Upgrade| Generating response file for autoupgrade] ***********************************************************
changed: [ansible_db]

TASK [db_upgrade : Autoupgrade | Analyze]
*****************************************************************************************************************
changed: [ansible_db]

TASK [db_upgrade : debug]
*****************************************************************************************************************
ok: [ansible_db] => {
   "analyze.stdout_lines": [
      "AutoUpgrade tool launched with default options",
      "Processing config file ...",
      "+--------------------------------+",
      "| Starting AutoUpgrade execution |",
      "+--------------------------------+",
      "1 databases will be analyzed",
      "Job 100 completed",
      "------------------ Final Summary --------------------",
      "Number of databases           [ 1 ]",
      "",
      "Jobs finished successfully     [1]",
      "Jobs failed                   [0]",
      "Jobs pending                  [0]",
      "------------- JOBS FINISHED SUCCESSFULLY -------------",
      "Job 100 for podbac"
   ]
}

TASK [db_upgrade : debug]
*****************************************************************************************************************
ok: [ansible_db] => {
   "msg": "Review the specific job number logfiles of Autoupgrade analyze task under /u02/base/podba_upgrade/autoupgrade_logs/podbac and take
appropriate action before continuing. Fix any errors and follow the recommendations reported in the logs and rerun this playbook, once all the errors are
fixed proceed with the upgrade using --tags deploy."
}

TASK [db_upgrade : Checking Source database name in /etc/oratab file] **************************************************************
skipping: [ansible_db]

TASK [db_upgrade : fail]
*****************************************************************************************************************
skipping: [ansible_db]

TASK [db_upgrade : Checking Target Container database name in /etc/oratab file] ****************************************************
skipping: [ansible_db]

TASK [db_upgrade : fail]
*****************************************************************************************************************
skipping: [ansible_db]

TASK [db_upgrade : Checking the status of Source DB] *******************************************************************************
skipping: [ansible_db]

TASK [db_upgrade : fail]
*****************************************************************************************************************
skipping: [ansible_db]

TASK [db_upgrade : Checking the status of Container DB] ****************************************************************************
```

```
skipping: [ansible_db]

TASK [db_upgrade : fail]
*********************************************************************************************************
skipping: [ansible_db]

TASK [db_upgrade : Creating Autoupgrade Stage directory] ***********************************************************************
skipping: [ansible_db]

TASK [db_upgrade : DB Upgrade| Generating response file for autoupgrade] ******************************************************
skipping: [ansible_db]

TASK [db_upgrade : Autoupgrade | Analyze]
*****************************************************************************************
skipping: [ansible_db]

TASK [db_upgrade : debug]
*********************************************************************************************************
skipping: [ansible_db]

TASK [db_upgrade : debug]
*********************************************************************************************************
skipping: [ansible_db]

PLAY RECAP
*********************************************************************************************************************************
ansible_db             : ok=55   changed=30   unreachable=0    failed=0    skipped=25   rescued=0    ignored=0
```

SIHA 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
```

Output from Step 5(b):

```
[ansible@x134vm236 playbooks]$ ansible-playbook upgrade_si.yml -i inventory.yml --ask-vault-pass --tags deploy
Vault password:
[WARNING]: running playbook inside collection ibm.power_aix_oracle_dba

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

TASK [db_upgrade : Autoupgrade | Deploy]
*****************************************************************************************
    "Total jobs 1",
    "",
    "Job 100 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/base/podba_upgrade/autoupgrade/cfgtoollogs/upgrade/auto/status/status.html",
    "/u02/base/podba_upgrade/autoupgrade/cfgtoollogs/upgrade/auto/status/status.log"
  ]
}

TASK [db_upgrade : Autoupgrade | Deploy]
*****************************************************************************************
skipping: [ansible_db]

TASK [db_upgrade : debug]
*********************************************************************************************************
skipping: [ansible_db]

PLAY RECAP
*********************************************************************************************************************************
rac91                  : ok=3    changed=1    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
```