Oracle Single Instance Dataguard Implementation using Ansible PowerODBA

1. Introduction

This document provides a comprehensive guide on automating the Data Guard setup of Oracle Single Instance database running on IBM Power AIX operating system supports both JFS and Oracle ASM. This have been tested with 19.25 & 19.26 oracle version. Also tested playbook using Ansible Automation Platform 2, it includes the architecture, prerequisites, limitations, playbook details, variables explanation and execution steps. Using Ansible, this solution streamlines the physical standby creation.

2. Versions

1.3.0

➢ Dataguard support is available on PowerODBA

3. Architecture Diagram

Diagram representation of the primary and standby database setup with Data Guard, Ansible control node, and network communication.

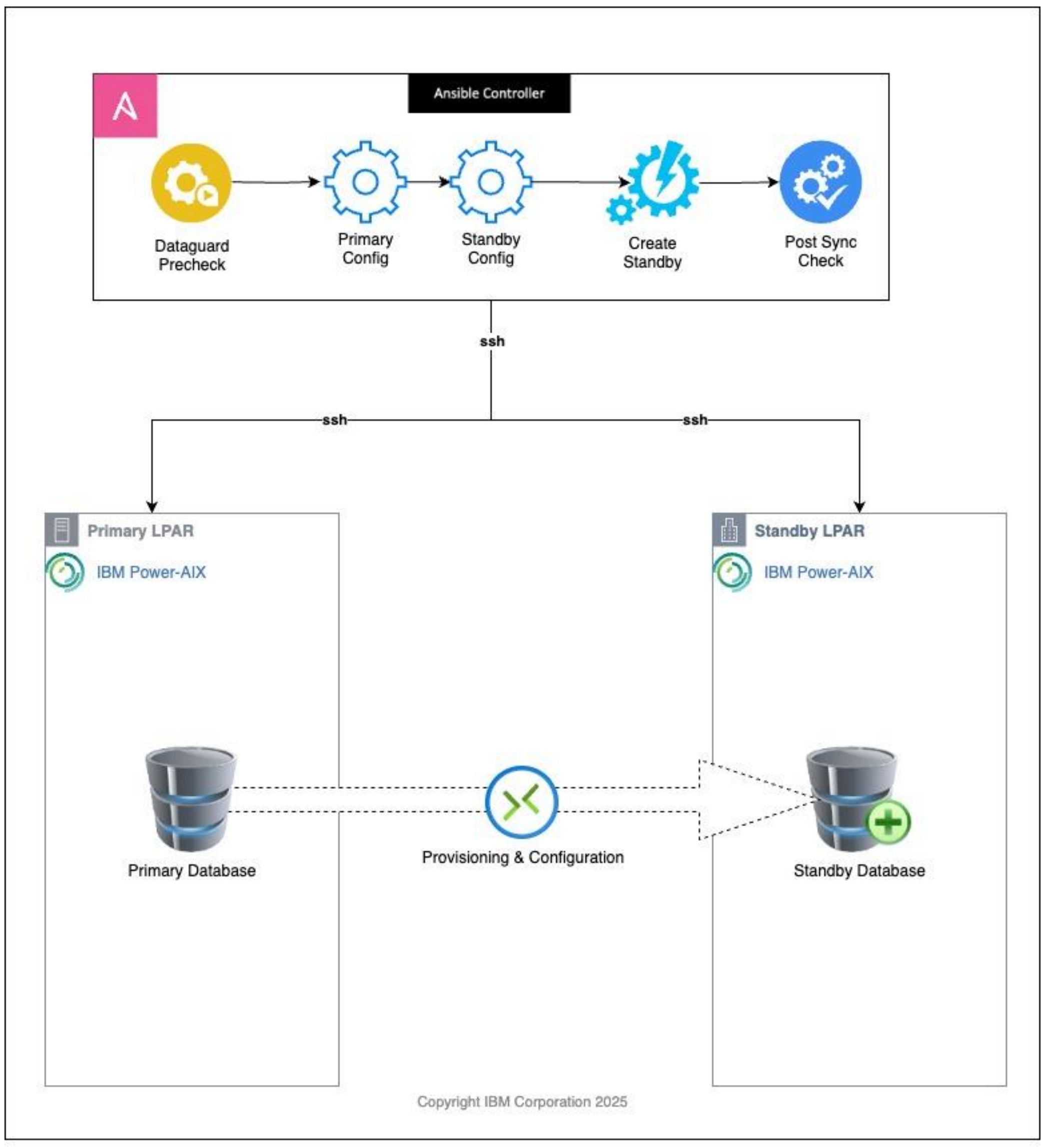

Fig: 1 Oracle Single Instance Dataguard Implementation using Ansible PowerODBA Collection

4. Prerequisites

This playbook assumes the following:

- Oracle 19c Software installed on both nodes primary and standby
- Ansible installed on the control node
- Passwordless SSH setup between control node and both database servers
- Requires connectivity between the primary and standby databases
- Sufficient storage and memory resources on both sites
- ASM disk groups or JFS file systems configured
- On the standby server with ASM, set up passwordless SSH (SSH equivalence) from the Oracle user to the Grid user — only if they are different users.
- TNS Alias for primary database should be already present at primary site
- Update the backup_location in vars if different in both primary and standby before invoking the respective play

To get started with Ansible refer

https://docs.ansible.com/ansible/latest/user_guide/intro_getting_started.html

To get started with Oracle Database on AIX refer

https://docs.oracle.com/en/database/oracle/oracle-database/19/axdbi/index.html

To get started with AIX refer

https://www.ibm.com/support/knowledgecenter/ssw_aix_72/navigation/welcome.html

5. Ansible Playbook Setup and Execution Steps

Below section provides the detail steps that are need to be followed for running the playbook.

5.1 Install Ansible Controller on your preferred operation system. We have installed and tested Ansible Controller on x86 server and Linux-on-Power server

Refer : https://docs.ansible.com/ansible/latest/installation_guide/intro_installation.html

We have created a user "ansible" on LoP server and considered /home/ansible as working directory.

5.2 Setup ssh Equivalence with managed host(AIX) (Primary and Standby) server

If this is a first time using ssh, then you probably haven't created your ssh keys. To check go to ~/.ssh and see if id_rsa file exists. If not you must create the ssh keys.

To create the ssh keys, run the following:

➢ `ssh-keygen`

Next copy the keys to the managed host.

➢ `ssh-copy-id root@<managed_host>`

Eg: ssh-copy-id root@aixhost

Check/Verify ssh connectivity

➢ `ssh root@aixhost`

5.3 Download playbook from Galaxy

Download the Oracle Dataguard ansible collection from ansible galaxy or github.

https://galaxy.ansible.com/ui/repo/published/ibm/power_aix_oracle
https://github.com/IBM/ansible-power-aix-oracle

To download from github you can follow below steps

```
$ cd <working directory>
$ git clone https://github.com/IBM/ansible-power-aix-oracle.git
$ cd ansible-power-aix-oracle
```

Download the power_aix_oracle tarball from ansible-galaxy and extract it to some directory for offline use or you can run the ansible-galaxy installation command:

```
ansible-galaxy collection install ibm.power_aix_oracle
```

The above command will install power_aix_oracle collection at location ~/.ansible/collections
For more information regarding ansible collections refer below link

https://docs.ansible.com/ansible/latest/user_guide/collections_using.html

6. Ansible Playbook Description

6.1 Playbook Files

1. **Path to the collection**: $*ansible-collection-install-dir*/ibm/ansible-power-aix-oracle-dba.
2. **dataguard_rman_backup_playbook.yml**: This is the *playbook* file which is responsible for taking RMAN backup of primary database. The file is under - **"*ansible-collection-install-dir*"/ibm/ansible-power-aix-oracle-dba/playbooks**.
3. **dataguard_playbook.yml**: This is the *playbook* file which is responsible for provisioning and configuring oracle physical standby database based on primary database config by calling the respective roles. The file is under - **"*ansible-collection-install-dir*"/ibm/ansible-power-aix-oracle-dba/playbooks**.
4. **inventory.yml**: This file is provided in the collection which contain all the managed hosts (Primary and Standby) 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.
5. **dataguard_vars.yml**: This file contains all the variables required to take RMAN backup and create the standby database. It is under is under - **"*ansible-collection-installdir*"/ibm/ansible-power-aix-oracle-dba/playbooks/vars/dataguard**. Specification of each variable is provided in this file itself.
6. **vault.yml**: The sys user password of primary database must be mentioned in this file, this file is in **"*ansible-collection-installdir*"/ibm/ansible-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*

6.2 Ansible Playbook Structure

The playbook is organized into five roles for ease of maintenance and reusability:

1. **dataguard_precheck**:
   - Pre-config checks to validate current primary database configurations
   - Pre-config checks to validate current standby server
2. **primary_config**:
   - Configure Dataguard parameters in primary database
   - Enables FORCE LOGGING
   - Creates the Standby Redo Logs
3. **standby_config**:
   - Configure Dataguard parameters in standby database
4. **create_standby:**
   - Creates standby using RMAN duplicate or RMAN Backup
   - Starts the standby database in managed recovery mode
5. **Post_sync_check:**
   - Checks Data Guard synchronization.
   - Ensures logs are applied on the standby

6.3 Ansible Playbook Roles

Detailed description of the actions performed by each Ansible role in the standby provisioning and configuring process:

dataguard_precheck
Performs the precheck required for 19c standby configuration
1. Checks if Dataguard setup was already done 2. Verifies network connectivity between primary and standby 3. Check Grid home Oracle DB home patch set 4. Checks the primary database is in ARCHIVELOG_MODE 5. Checks the standby site has enough space 6. Performs System Configuration Comparison between primary and standby and show warning 7. Validates the input variable in dataguard_vars.yml file

primary_config
Configures the primary database
1. Checks if Dataguard setup was already done 2. Run Dataguard Pre-config tasks for primary 3. Enable FORCE_LOGGING if not already 4. Enable standby file management 5. Create standby redo logs (SRL) 6. ADD TNS alias added for standby database tnsnames.ora 7. Modify the primary initialization parameter for dataguard on primary

standby_config
Configures the standby database
1. Checks if Dataguard setup was already done 2. Run Dataguard Pre-config tasks for standby 3. ADD TNS alias added for primary database tnsnames.ora 4. Create temporary listener for RMAN duplicate method only 5. Validates successful TNS connections between both sites 6. Set the standby initialization parameter 7. Fetch password and PFILE file from primary

create_standby
Provisioning of standby database
1. Checks if Dataguard setup was already done 2. Fetch DBID and control file location from primary database 3. Run Dataguard Restore Script on standby 4. Validate Dataguard restore Status and Run Restore Post-Processing Script on standby 5. Run Protection Mode Post-Processing on primary to MAX_PERFORMANCE 6. Remove temporary listener for RMAN duplicate method only

post_sync_check
Checks Data Guard synchronization
1. Validate Data Guard is in sync with primary or not 2. Compare sequence numbers between primary and standby 3. Attempt to sync by restarting redo logs shipping and restarting MRP 4. Displays the Data Guard Sync Status

7. Limitations

- This setup has been tested on Oracle 19c Single Instance running on AIX
- If GAP is more then take incremental backup and sync manually
- Having identical software and hardware configurations between the primary and standby sites is desirable for optimal performance
- Update default parameter primary_cpu_core/ primary_cpu_core in defaults if more than 4 parallelism is required for RMAN backup and restore
- Force logging is enabled by default if not already set on the primary
- Assumes all disk groups or file systems are already created with appropriate permission
- In this release, Data Guard operates in MAXIMUM PERFORMANCE mode only
- Multiple standby database is not supported in this release
- Always run the playbooks in a vnc viewer to avoid ssh timeouts.
- These playbooks will create three directory ansible in /tmp. It should NOT be removed until the dataguard setup completes otherwise it will compromise idempotency.
- Try this on a non-production environment first before using it on a Production environment.

8. Methods for Configuring Standby Database

8.1 RMAN Duplicate

1. Uses RMAN DUPLICATE TARGET DATABASE command
2. No need to take or transfer backup files manually
3. Supports both ASM and JFS file systems

8.2 RMAN Backup-Based

1. Take an RMAN backup of the primary database using ansible play or manually
2. Transfer backup files to the standby server manually
3. Restores and recovers the standby database
4. Supports both ASM and JFS file systems

8.3 Parameters for each methods

Methods	grid_asm_flag	with_backup
JFS2 RMAN Backup based	false	true
JFS2 RMAN ACTIVE Duplicate	false	false
ASM RMAN Backup	true	true
ASM RMAN ACTIVE Duplicate	true	false

9. Preparing to run the Dataguard playbook

9.1 Change directory to the galaxy_collection directory

```
$ cd ansible-power-aix-oracle
```

9.2 Update the inventory

```
[primary]
primary-db ansible_host=111.11.11.111

[standby]
standby-db ansible_host=111.11.11.111
```

9.3 Update the variable file

Please update the variables by referring Sample variables for selected methods are available at vars/sample_vars directory.

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

```
$ cat vars/vault.yml

asm_password: ########  # sys user password for Primary database

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

10. Execute the playbook

10.1 RMAN ACTIVE DUPLICATE

No prior backup needed , RMAN active duplicate will perform it online .

dataguard_playbook.yml:

```
# This playbook is used to configure dataguard for 19c.
# Before executing this playbook, please refer the documentation inside the docs directory.
#
# - hosts: all_nodes                       # Provide the name of the target lpar registered in ansible 
inventory.
#   remote_user: oracle                      # This needs to be run by "oracle" user.
#   gather_facts: False
#  vars_files:
#      - vars/dataguard/dataguard_vars.yml  # update all the required variables
#      - vars/vault.yml       # update the passwords

- name: Configure Data Guard for 19c
  hosts: all
  remote_user: "{{ db_oracle_user }}"
  gather_facts: false
  vars_files:
    - vars/dataguard/dataguard_vars.yml
    - vars/vault.yml

  roles:
    - role: dataguard_precheck
      tags: dataguard_precheck

    - role: primary_config
      tags: primary_config

    - role: standby_config
      tags: standby_config

    - role: create_standby
      tags: create_standby

    - role: post_sync_check
      tags: post_sync_check
```

Provision and configure standby: This command will perform end to end setup of standby database.

```
ansible-playbook dataguard_play.yml -i inventory.yml --ask-vault-pass --tags
dataguard_precheck,primary_config, standby_config,create_standby,post_sync_check
                                    or
ansible-playbook dataguard_playbook.yml -i inventory.yml --ask-vault-pass
```

Check GAP: This command will check and attempt to sync if gap is minimal and can resolved within three minutes

```
ansible-playbook dataguard_playbook.yml -i inventory.yml --ask-vault-pass --tags post_sync_check
```

10.2 RMAN BACKUP BASED

Take RMAN backup of primary database using below ansible playbook and manually transfer the backup at standby site.

dataguard_rman_backup_playbook.yml:

```
# This playbook is used to secure backup to create dataguard for 19c.
# Before executing this playbook, please refer the documentation inside the docs directory.
#
# - hosts: all_nodes                            # Provide the name of the target lpar registered in
ansible inventory.
#  remote_user: oracle                     # This needs to be run by "oracle" user.
#  gather_facts: False
#  vars_files:
#      - vars/dataguard/dataguard_vars.yml  # update all the required variables
#      - vars/vault.yml       # update the passwords

- name: Configure Data Guard for 19c
  hosts: all
  remote_user: "{{ db_oracle_user }}"
  gather_facts: false
  vars_files:
    - vars/dataguard/dataguard_vars.yml
    - vars/vault.yml

  roles:
    - role: backup_primary_db
      tags: backup_primary_db
```

RMAN backup: Initiate RMAN backup for primary. This command will perform the RMAN backup at primary database.

```
ansible-playbook rman_backup_playbook.yml -i inventory.yml --ask-vault-pass
```

Note: Please transfer the backup to standby server.

Provision and configure standby: This command will perform end to end setup of standby database.

```
ansible-playbook dataguard_play.yml -i inventory.yml --ask-vault-pass --tags
dataguard_precheck,primary_config, standby_config,create_standby,post_sync_check
                                    or
ansible-playbook dataguard_playbook.yml -i inventory.yml --ask-vault-pass
```

Check GAP: This command will check and attempt to sync if gap is minimal and can resolved within three minutes

```
ansible-playbook dataguard_playbook.yml -i inventory.yml --ask-vault-pass --tags post_sync_check
```

11. Troubleshooting

If you see any failures during the playbook execution, to get more details regarding the failure try using -vvv option

```
$ ansible-playbook dataguard_playbook.yml --tags “preconfig” -vvv
```

You can create a github issue and our team will look into this

https://github.com/IBM/ansible-power-aix-oracle/issues

Sample Execution logs: In the following example, we are going to setup standby database using RMAN ACTIVE DUPLICATE method for ASM

```
ansible-playbook dataguard_playbook.yml -i inventory

PLAY [Configure Data Guard for 19c]
*************************************************************************************************************************************************
********

TASK [dataguard_precheck : Run initialization tasks]
****************************************************************************************************************************************
included: /var/lib/awx/projects/_161__dg_setup/roles/dataguard_precheck/tasks/../../dataguard_precheck/tasks/init.yml for primary-db, standby-db

TASK [dataguard_precheck : Set fact init_done]
**********************************************************************************************************************************************
ok: [primary-db]
ok: [standby-db]

TASK [dataguard_precheck : Set fact scripts dir]
********************************************************************************************************************************************
ok: [primary-db]
ok: [standby-db]

TASK [dataguard_precheck : Set fact done dir]
***********************************************************************************************************************************************
ok: [primary-db]
ok: [standby-db]

TASK [dataguard_precheck : Create work dir]
*************************************************************************************************************************************************
ok: [standby-db]
changed: [primary-db]

TASK [dataguard_precheck : Create scripts dir]
**********************************************************************************************************************************************
changed: [standby-db]
changed: [primary-db]

TASK [dataguard_precheck : Create done dir]
*************************************************************************************************************************************************
changed: [standby-db]
changed: [primary-db]

TASK [dataguard_precheck : Checking if Dataguard setup was already done]
********************************************************************************************************************
skipping: [standby-db]
ok: [primary-db]

TASK [dataguard_precheck : Exit if Dataguard setup was already done]
************************************************************************************************************************
skipping: [primary-db]

TASK [dataguard_precheck : End play if Dataguard setup Task was done]
***********************************************************************************************************************
skipping: [primary-db]

TASK [dataguard_precheck : Verify primary can ping standby host]
****************************************************************************************************************************
skipping: [standby-db]
ok: [primary-db]

TASK [dataguard_precheck : Fail if primary cannot reach standby]
****************************************************************************************************************************
skipping: [primary-db]
skipping: [standby-db]

TASK [dataguard_precheck : Verify standby can ping primary host]
****************************************************************************************************************************
skipping: [primary-db]
ok: [standby-db]

TASK [dataguard_precheck : Fail if standby cannot reach primary]
****************************************************************************************************************************
skipping: [primary-db]
skipping: [standby-db]

TASK [dataguard_precheck : Get release patch details from site]
*****************************************************************************************************************************
ok: [primary-db]
ok: [standby-db]

TASK [dataguard_precheck : Set patch level and patches facts]
*******************************************************************************************************************************
ok: [primary-db]
ok: [standby-db]

TASK [dataguard_precheck : Set standby patch details for comparison (on standby only)]
******************************************************************************************************
skipping: [primary-db]
ok: [standby-db]

TASK [dataguard_precheck : Set primary patch details for comparison (on primary only)]
******************************************************************************************************
ok: [primary-db]
skipping: [standby-db]

TASK [dataguard_precheck : Fail if patch levels or patches do not match between primary and standby for Oracle Clusterware]
*****************************************************************
skipping: [primary-db]
skipping: [standby-db]

TASK [dataguard_precheck : Get applied patch from Oracle DB home]
****************************************************************************************************************************
```

```
ok: [standby-db]
ok: [primary-db]

TASK [dataguard_precheck : Set patch fact]
**************************************************************************************************************************************************
*
ok: [primary-db]
ok: [standby-db]

TASK [dataguard_precheck : Set standby patch details for comparison (on standby only)]
*******************************************************************************************************
skipping: [primary-db]
ok: [standby-db]

TASK [dataguard_precheck : Set primary patch details for comparison (on primary only)]
*******************************************************************************************************
ok: [primary-db]
skipping: [standby-db]

TASK [dataguard_precheck : Fail if patch numbers do not match between primary and standby]
***************************************************************************************************
skipping: [primary-db]
skipping: [standby-db]

TASK [dataguard_precheck : Checking if Dataguard setup was already done]
*********************************************************************************************************************
skipping: [standby-db]
ok: [primary-db]

TASK [dataguard_precheck : Copy Dataguard Pre-check script for primary]
**********************************************************************************************************************
skipping: [standby-db]
changed: [primary-db]

TASK [dataguard_precheck : Execute Dataguard Pre-check script on primary]
********************************************************************************************************************
skipping: [standby-db]
ok: [primary-db]

TASK [dataguard_precheck : Debug Pre-check Output for primary database]
**********************************************************************************************************************
ok: [primary-db] => {
    "msg": [
        "ARCHIVELOG_MODE:ARCHIVELOG",
        "DATABASE_SIZE:1.43",
        "DISK_COUNT:1",
        "REDO_SIZES:209715200,209715200,209715200",
        "MULTIPLEX_CHECK:1",
        "MULTIPLEX_CHECK:1",
        "MULTIPLEX_CHECK:1",
        "All pre-checks passed for primary database successfully!"
    ]
}
skipping: [standby-db]

TASK [dataguard_precheck : Extract DATABASE_SIZE from log]
***********************************************************************************************************************************
skipping: [standby-db]
ok: [primary-db]

TASK [dataguard_precheck : Set DATABASE_SIZE fact on primary]
********************************************************************************************************************************
skipping: [standby-db]
ok: [primary-db]

TASK [dataguard_precheck : Fetch DATABASE_SIZE fact from primary to standby]
*****************************************************************************************************************
skipping: [primary-db]
ok: [standby-db]

TASK [dataguard_precheck : Checking if Dataguard setup was already done]
*********************************************************************************************************************
skipping: [primary-db]
ok: [standby-db]

TASK [dataguard_precheck : Copy Dataguard Pre-check script for standby]
**********************************************************************************************************************
skipping: [primary-db]
changed: [standby-db]

TASK [dataguard_precheck : Allow grid user to access Ansible temporary directory]
************************************************************************************************************
skipping: [primary-db]
ok: [standby-db]

TASK [dataguard_precheck : Execute Dataguard Pre-check script on standby]
********************************************************************************************************************
skipping: [primary-db]
ok: [standby-db]

TASK [dataguard_precheck : Debug Pre-check Output for standby database]
**********************************************************************************************************************
skipping: [primary-db]
ok: [standby-db] => {
    "msg": [
        "Checking existence of Disk Group or Filesystem: +DATAS",
        "disk_count:1",
        "Disk group +DATAS exists.",
        "Checking existence of Disk Group or Filesystem: +DATAS",
        "disk_count:1",
        "Disk group +DATAS exists.",
        "Checking existence of Disk Group or Filesystem: +DATAS",
        "disk_count:1",
        "Disk group +DATAS exists.",
        "Checking available space for +DATAS...",
        "available_space:95.05",
        "Available space in +DATAS: 95.05G",
        "Required space: 1.43G",
        "Sufficient space available in +DATAS.",
        "Available port: 1522",
        "All checks completed successfully.",
        "Pre-checks passed for standby database!"
    ]
}

TASK [dataguard_precheck : Execute Dataguard Pre-check script on standby without ssh]
********************************************************************************************************
skipping: [primary-db]
skipping: [standby-db]
```

```
TASK [dataguard_precheck : Debug Pre-check Output for standby database]
*********************************************************************************************************************
skipping: [primary-db]
skipping: [standby-db]

TASK [dataguard_precheck : Gather system information (CPU, Memory) on all nodes]
************************************************************************************************************
ok: [standby-db]
ok: [primary-db]

TASK [dataguard_precheck : Set system facts for comparison]
*********************************************************************************************************************************
ok: [primary-db]
ok: [standby-db]

TASK [dataguard_precheck : Fetch primary system configuration on standby]
*******************************************************************************************************************
skipping: [primary-db]
ok: [standby-db]

TASK [dataguard_precheck : Show warning if system configurations do not match]
**************************************************************************************************************
skipping: [primary-db]
skipping: [standby-db]

TASK [primary_config : Run initialization tasks]
********************************************************************************************************************************************
skipping: [primary-db]
skipping: [standby-db]

TASK [primary_config : Checking if Dataguard setup was already done]
************************************************************************************************************************
skipping: [standby-db]
ok: [primary-db]

TASK [primary_config : Exit if Dataguard setup was already done]
****************************************************************************************************************************
skipping: [primary-db]

TASK [primary_config : End play if Dataguard setup Task was done]
***************************************************************************************************************************
skipping: [primary-db]

TASK [primary_config : Checking if Primary pre-configure was already done]
******************************************************************************************************************
skipping: [standby-db]
ok: [primary-db]

TASK [primary_config : Fetch available_port from standby]
***********************************************************************************************************************************
skipping: [primary-db]
ok: [standby-db]

TASK [primary_config : Set available_port as fact on standby]
*******************************************************************************************************************************
skipping: [primary-db]
ok: [standby-db]

TASK [primary_config : Share available_port from standby to primary]
************************************************************************************************************************
skipping: [standby-db]
ok: [primary-db]

TASK [primary_config : Copy Dataguard Pre-config script for primary]
************************************************************************************************************************
skipping: [standby-db]
skipping: [primary-db]

TASK [primary_config : Copy Dataguard Pre-config script for primary with available_port]
****************************************************************************************************
skipping: [standby-db]
changed: [primary-db]

TASK [primary_config : Execute Dataguard Pre-config script on primary]
**********************************************************************************************************************
skipping: [standby-db]
ok: [primary-db]

TASK [primary_config : Debug Pre-config Output for primary database]
************************************************************************************************************************
ok: [primary-db] => {
    "msg": [
        "DB_LOGGING:YES",
        "STDBY_FILE_MANAGEMENT:AUTO",
        "REDO_COUNT:3,MAX_GROUP:3",
        "REDO_LOCATION:+DATA/ORCL/ONLINELOG/group_1.258.1196268193,REDO_SIZE:209715200",
        "SRL_COUNT:0",
        "SPFILE_CHECK:1",
        "Executed: ALTER DATABASE ADD STANDBY LOGFILE GROUP 4 ('+DATA') SIZE 209715200",
        "Executed: ALTER DATABASE ADD STANDBY LOGFILE GROUP 5 ('+DATA') SIZE 209715200",
        "Executed: ALTER DATABASE ADD STANDBY LOGFILE GROUP 6 ('+DATA') SIZE 209715200",
        "Executed: ALTER DATABASE ADD STANDBY LOGFILE GROUP 7 ('+DATA') SIZE 209715200",
        "",
        "PL/SQL procedure successfully completed.",
        "",
        "Database is using SPFILE. Creating PFILE from SPFILE...",
        "",
        "File created.",
        "",
        "TNS alias added for standby database tnsnames.ora.",
        "",
        "System altered.",
        "",
        "",
        "System altered.",
        "",
        "",
        "System altered.",
        "",
        "",
        "System altered.",
        "",
        "",
        "System altered.",
        "",
        "",
        "System altered.",
        "",
        "",
        "System altered.",
        "",
        "DB initialization parameter set completed successfully.",
        "Standby redo logs added, PFILE copied, TNS entries updated and init params modified for dataguard on primary",
        "All pre-config performed for primary database successfully!"
    ]
}
skipping: [standby-db]

TASK [primary_config : Fetch password and PFILE file from primary]
****************************************************************************************************************************
skipping: [standby-db] => (item={'src': '/tmp/ansible/scripts/primary.pfile', 'dest': 'standby.pfile'})
skipping: [standby-db] => (item={'src': '/u01/product/19.3.0.0.0/database/dbs/orapworcl', 'dest': 'orapworcl'})
skipping: [standby-db]
ok: [primary-db] => (item={'src': '/tmp/ansible/scripts/primary.pfile', 'dest': 'standby.pfile'})
ok: [primary-db] => (item={'src': '/u01/product/19.3.0.0.0/database/dbs/orapworcl', 'dest': 'orapworcl'})

TASK [primary_config : Copy fetched files from control node to standby]
***********************************************************************************************************************
ok: [primary-db -> standby-db(129.40.76.242)] => (item={'src': 'orapworcl', 'mode': '0640'})
ok: [primary-db -> standby-db(129.40.76.242)] => (item={'src': 'standby.pfile', 'mode': '0644'})

TASK [standby_config : Run initialization tasks]
**********************************************************************************************************************************************
skipping: [primary-db]
skipping: [standby-db]

TASK [standby_config : Checking if Dataguard setup was already done]
**************************************************************************************************************************
skipping: [standby-db]
ok: [primary-db]

TASK [standby_config : Exit if Dataguard setup was already done]
******************************************************************************************************************************
skipping: [primary-db]

TASK [standby_config : End play if Dataguard setup Task was done]
*****************************************************************************************************************************
skipping: [primary-db]

TASK [standby_config : Checking if Standby configuration was already done]
********************************************************************************************************************
skipping: [primary-db]
ok: [standby-db]

TASK [standby_config : Create necessary directories for standby configuration]
****************************************************************************************************************
skipping: [primary-db] => (item=adump)
skipping: [primary-db] => (item=dpump)
skipping: [primary-db]
ok: [standby-db] => (item=adump)
ok: [standby-db] => (item=dpump)

TASK [standby_config : Fetch available_port from standby]
*************************************************************************************************************************************
skipping: [primary-db]
ok: [standby-db]

TASK [standby_config : Set available_port as fact on standby]
*********************************************************************************************************************************
skipping: [primary-db]
ok: [standby-db]

TASK [standby_config : Copy Dataguard Pre-config script for standby]
**************************************************************************************************************************
skipping: [primary-db]
skipping: [standby-db]

TASK [standby_config : Copy Dataguard Pre-config script for standby with available_port]
******************************************************************************************************
skipping: [primary-db]
changed: [standby-db]

TASK [standby_config : Execute Dataguard config script on standby]
****************************************************************************************************************************
skipping: [primary-db]
ok: [standby-db]

TASK [standby_config : Debug config Output for standby database]
******************************************************************************************************************************
skipping: [primary-db]
ok: [standby-db] => {
    "msg": [
        "TNS alias added for standby database tnsnames.ora.",
        "TNS alias added for primary database tnsnames.ora.",
        "tnsnames.ora not found. Creating new file...",
        "File /u01/product/19.3.0.0.0/database/network/admin/orcls_rman_listener/listener.ora does not exist. Creating it...",
        "Updating /u01/product/19.3.0.0.0/database/network/admin/orcls_rman_listener/listener.ora...",
        "RMAN listener entry added successfully.",
        "Listener start successful",
        "Listener is running successfully",
        "Instance 'orcl' is in UNKNOWN state!",
        "TNS alias added for standby database tnsnames.ora.",
        "TNS alias added for primary database tnsnames.ora.",
        "Standby PFILE, PWD file copied, and TNS entries updated.",
        "All pre-config performed for standby database successfully!"
    ]
}

TASK [standby_config : Check connectivity to primary database from standby site]
**************************************************************************************************************
skipping: [primary-db]
ok: [standby-db]

TASK [standby_config : Validate primary database connectivity from standby site]
**************************************************************************************************************
skipping: [primary-db]
skipping: [standby-db]

TASK [standby_config : Confirm successful TNS connections from standby site]
******************************************************************************************************************
skipping: [primary-db]
ok: [standby-db] => {
    "msg": "All good for TNS connections from standby!"
}

TASK [standby_config : Check connectivity to standby database from primary site]
**************************************************************************************************************
skipping: [standby-db]
ok: [primary-db]
```

```
TASK [standby_config : Validate standby database connectivity from primary site]
*************************************************************************************************************
skipping: [standby-db]
skipping: [primary-db]

TASK [standby_config : Confirm successful TNS connections from primary site]
******************************************************************************************************************
skipping: [standby-db]
ok: [primary-db] => {
    "msg": "All good for TNS connections from primary!"
}

TASK [create_standby : Run initialization tasks]
**********************************************************************************************************************************************
skipping: [primary-db]
skipping: [standby-db]

TASK [create_standby : Checking if Dataguard setup was already done]
**************************************************************************************************************************
skipping: [standby-db]
ok: [primary-db]

TASK [create_standby : Exit if Dataguard setup was already done]
******************************************************************************************************************************
skipping: [primary-db]

TASK [create_standby : End play if Dataguard setup Task was done]
*****************************************************************************************************************************
skipping: [primary-db]

TASK [create_standby : Checking if Dataguard restore was already done]
************************************************************************************************************************
skipping: [primary-db]
ok: [standby-db]

TASK [create_standby : Copy DBID and control file location script from primary database]
*****************************************************************************************************
skipping: [standby-db]
changed: [primary-db]

TASK [create_standby : Execute DBID and control file location script on primary database]
****************************************************************************************************
skipping: [standby-db]
ok: [primary-db]

TASK [create_standby : Fetch DBID file content]
***********************************************************************************************************************************************
skipping: [standby-db]
ok: [primary-db]

TASK [create_standby : Set DBID as facts]
***************************************************************************************************************************************************
**
skipping: [standby-db]
ok: [primary-db]

TASK [create_standby : Fetch DBID facts from primary to standby]
******************************************************************************************************************************
skipping: [primary-db]
ok: [standby-db]

TASK [create_standby : Copy RMAN restore script for standby]
**********************************************************************************************************************************
skipping: [primary-db]
changed: [standby-db]

TASK [create_standby : Execute restore script for standby]
************************************************************************************************************************************
skipping: [primary-db]
ok: [standby-db]

TASK [create_standby : Debug restore Output for standby database]
*****************************************************************************************************************************
skipping: [primary-db]
ok: [standby-db] => {
    "msg": [
        "Starting standby database...",
        "ORACLE instance started.",
        "",
        "Total System Global Area  583000016 bytes",
        "Fixed Size\t\t    9101264 bytes",
        "Variable Size\t\t  297795584 bytes",
        "Database Buffers\t  268435456 bytes",
        "Redo Buffers\t\t    7667712 bytes",
        "",
        "Recovery Manager: Release 19.0.0.0.0 - Production on Fri Apr 4 13:46:20 2025",
        "Version 19.26.0.0.0",
        "",
        "Copyright (c) 1982, 2019, Oracle and/or its affiliates.  All rights reserved.",
        "",
        "connected to target database: ORCL (DBID=1724048925)",
        "connected to auxiliary database: ORCL (not mounted)",
        "",
        "RMAN> RUN {",
        "2>     ALLOCATE CHANNEL prmy1 DEVICE TYPE DISK ;",
        "3>     ALLOCATE CHANNEL prmy2 DEVICE TYPE DISK ;",
        "4>     ALLOCATE CHANNEL prmy3 DEVICE TYPE DISK ;",
        "5>     ALLOCATE CHANNEL prmy4 DEVICE TYPE DISK ;",
        "6>     ALLOCATE auxiliary CHANNEL stby TYPE DISK;",
        "7> ",
        "8>  duplicate target database for standby from active database NOFILENAMECHECK",
        "9>  spfile",
        "10>    parameter_value_convert 'orcl','orcls'",
        "11>    set db_name='orcl'",
        "12>    set db_create_file_dest='+DATAS'",
        "13>    set db_unique_name='orcls'",
        "14>    set db_file_name_convert='+DATA/orcl','+DATAS/orcls'",
        "15>    set log_file_name_convert='+DATA/orcl','+DATAS/orcls'",
        "16>    set control_files='+DATAS/orcls_control01.ctl'",
        "17>    set log_archive_max_processes='5'",
        "18>    set fal_client='orcls'",
        "19>    set fal_server='orcl'",
        "20>    set standby_file_management='MANUAL'",
        "21>    set log_archive_config='dg_config=(orcl,orcls)'",
        "22>    set log_archive_dest_2='service=orcl ASYNC NOAFFIRM valid_for=(ONLINE_LOGFILE,PRIMARY_ROLE) db_unique_name=orcl'",
        "23>    set db_create_online_log_dest_1='+DATAS'",
        "24>    set log_archive_dest_1='LOCATION=USE_DB_RECOVERY_FILE_DEST'",
        "25>    set db_recovery_file_dest='+DATAS'",
        "26>    set db_recovery_file_dest_size='10G'",
        "27>  ;",
        "28> }",
        "29> EXIT;",
        "using target database control file instead of recovery catalog",
        "allocated channel: prmy1",
        "channel prmy1: SID=292 device type=DISK",
        "",
        "allocated channel: prmy2",
        "channel prmy2: SID=574 device type=DISK",
        "",
        "allocated channel: prmy3",
        "channel prmy3: SID=858 device type=DISK",
        "",
        "allocated channel: prmy4",
        "channel prmy4: SID=1140 device type=DISK",
        "",
        "allocated channel: stby",
        "channel stby: SID=87 device type=DISK",
        "",
        "Starting Duplicate Db at 04-APR-25",
        "",
        "contents of Memory Script:",
        "{",
        "   backup as copy reuse",
        "   passwordfile auxiliary format  '/u01/product/19.3.0.0.0/database/dbs/orapworcl'   targetfile ",
        " '+DATA/ORCL/PARAMETERFILE/spfile.266.1196269363' auxiliary format ",
        " '/u01/product/19.3.0.0.0/database/dbs/spfileorcl.ora'   ;",
        "   sql clone \"alter system set spfile= ''/u01/product/19.3.0.0.0/database/dbs/spfileorcl.ora''\";",
        "}",
        "executing Memory Script",
        "",
        "Starting backup at 04-APR-25",
        "Finished backup at 04-APR-25",
        "",
        "sql statement: alter system set spfile= ''/u01/product/19.3.0.0.0/database/dbs/spfileorcl.ora''",
        "",
        "contents of Memory Script:",
        "{",
        "   sql clone \"alter system set  audit_file_dest = ",
        " ''/u01/base/admin/orcls/adump'' comment=",
        " '''' scope=spfile\";",
        "   sql clone \"alter system set  dispatchers = ",
        " ''(PROTOCOL=TCP) (SERVICE=orclsXDB)'' comment=",
        " '''' scope=spfile\";",
        "   sql clone \"alter system set  db_name = ",
        " ''orcl'' comment=",
        " '''' scope=spfile\";",
        "   sql clone \"alter system set  db_create_file_dest = ",
        " ''+DATAS'' comment=",
        " '''' scope=spfile\";",
        "   sql clone \"alter system set  db_unique_name = ",
        " ''orcls'' comment=",
        " '''' scope=spfile\";",
        "   sql clone \"alter system set  db_file_name_convert = ",
        " ''+DATA/orcl'', ''+DATAS/orcls'' comment=",
        " '''' scope=spfile\";",
        "   sql clone \"alter system set  log_file_name_convert = ",
        " ''+DATA/orcl'', ''+DATAS/orcls'' comment=",
        " '''' scope=spfile\";",
        "   sql clone \"alter system set  control_files = ",
        " ''+DATAS/orcls_control01.ctl'' comment=",
        " '''' scope=spfile\";",
        "   sql clone \"alter system set  log_archive_max_processes = ",
        " 5 comment=",
        " '''' scope=spfile\";",
        "   sql clone \"alter system set  fal_client = ",
        " ''orcls'' comment=",
        " '''' scope=spfile\";",
        "   sql clone \"alter system set  fal_server = ",
        " ''orcl'' comment=",
        " '''' scope=spfile\";",
        "   sql clone \"alter system set  standby_file_management = ",
        " ''MANUAL'' comment=",
        " '''' scope=spfile\";",
        "   sql clone \"alter system set  log_archive_config = ",
        " ''dg_config=(orcl,orcls)'' comment=",
        " '''' scope=spfile\";",
        "   sql clone \"alter system set  log_archive_dest_2 = ",
        " ''service=orcl ASYNC NOAFFIRM valid_for=(ONLINE_LOGFILE,PRIMARY_ROLE) db_unique_name=orcl'' comment=",
        " '''' scope=spfile\";",
        "   sql clone \"alter system set  db_create_online_log_dest_1 = ",
        " ''+DATAS'' comment=",
        " '''' scope=spfile\";",
        "   sql clone \"alter system set  log_archive_dest_1 = ",
        " ''LOCATION=USE_DB_RECOVERY_FILE_DEST'' comment=",
        " '''' scope=spfile\";",
        "   sql clone \"alter system set  db_recovery_file_dest = ",
        " ''+DATAS'' comment=",
        " '''' scope=spfile\";",
        "   sql clone \"alter system set  db_recovery_file_dest_size = ",
        " 10G comment=",
        " '''' scope=spfile\";",
        "   shutdown clone immediate;",
        "   startup clone nomount;",
        "}",
        "executing Memory Script",
        "",
        "sql statement: alter system set  audit_file_dest =  ''/u01/base/admin/orcls/adump'' comment= '''' scope=spfile",
        "",
        "sql statement: alter system set  dispatchers =  ''(PROTOCOL=TCP) (SERVICE=orclsXDB)'' comment= '''' scope=spfile",
        "",
        "sql statement: alter system set  db_name =  ''orcl'' comment= '''' scope=spfile",
        "",
        "sql statement: alter system set  db_create_file_dest =  ''+DATAS'' comment= '''' scope=spfile",
        "",
        "sql statement: alter system set  db_unique_name =  ''orcls'' comment= '''' scope=spfile",
        "",
        "sql statement: alter system set  db_file_name_convert =  ''+DATA/orcl'', ''+DATAS/orcls'' comment= '''' scope=spfile",
        "",
        "sql statement: alter system set  log_file_name_convert =  ''+DATA/orcl'', ''+DATAS/orcls'' comment= '''' scope=spfile",
        "",
        "sql statement: alter system set  control_files =  ''+DATAS/orcls_control01.ctl'' comment= '''' scope=spfile",
        "",
        "sql statement: alter system set  log_archive_max_processes =  5 comment= '''' scope=spfile",
        "",
        "sql statement: alter system set  fal_client =  ''orcls'' comment= '''' scope=spfile",
        "",
        "sql statement: alter system set  fal_server =  ''orcl'' comment= '''' scope=spfile",
        "",
        "sql statement: alter system set  standby_file_management =  ''MANUAL'' comment= '''' scope=spfile",
        "",
        "sql statement: alter system set  log_archive_config =  ''dg_config=(orcl,orcls)'' comment= '''' scope=spfile",
        "",
        "sql statement: alter system set  log_archive_dest_2 =  ''service=orcl ASYNC NOAFFIRM valid_for=(ONLINE_LOGFILE,PRIMARY_ROLE)
db_unique_name=orcl'' comment= '''' scope=spfile",
        "",
        "sql statement: alter system set  db_create_online_log_dest_1 =  ''+DATAS'' comment= '''' scope=spfile",
        "",
        "sql statement: alter system set  log_archive_dest_1 =  ''LOCATION=USE_DB_RECOVERY_FILE_DEST'' comment= '''' scope=spfile",
        "",
        "sql statement: alter system set  db_recovery_file_dest =  ''+DATAS'' comment= '''' scope=spfile",
        "",
        "sql statement: alter system set  db_recovery_file_dest_size =  10G comment= '''' scope=spfile",
        "",
        "Oracle instance shut down",
        "",
        "connected to auxiliary database (not started)",
        "Oracle instance started",
        "",
        "Total System Global Area   10334714016 bytes",
        "",
        "Fixed Size                    13842592 bytes",
        "Variable Size               1610612736 bytes",
        "Database Buffers            8690597888 bytes",
        "Redo Buffers                  19660800 bytes",
        "allocated channel: stby",
        "channel stby: SID=2551 device type=DISK",
        "",
        "contents of Memory Script:",
        "{",
        "   backup as copy current controlfile for standby auxiliary format  '+DATAS/orcls_control01.ctl';",
        "}",
        "executing Memory Script",
        "",
        "Starting backup at 04-APR-25",
        "channel prmy1: starting datafile copy",
        "copying standby control file",
        "output file name=/u01/product/19.3.0.0.0/database/dbs/snapcf_orcl.f tag=TAG20250404T134645",
        "channel prmy1: datafile copy complete, elapsed time: 00:00:03",
        "Finished backup at 04-APR-25",
        "",
        "contents of Memory Script:",
        "{",
        "   sql clone 'alter database mount standby database';",
        "}",
        "executing Memory Script",
        "",
        "sql statement: alter database mount standby database",
        "RMAN-05529: warning: DB_FILE_NAME_CONVERT resulted in invalid ASM names; names changed to disk group only.",
        "",
        "contents of Memory Script:",
        "{",
        "   set newname for tempfile  1 to ",
        " \"+DATAS\";",
        "   switch clone tempfile all;",
        "   set newname for datafile  1 to ",
        " \"+DATAS\";",
        "   set newname for datafile  2 to ",
        " \"+DATAS\";",
        "   set newname for datafile  3 to ",
        " \"+DATAS\";",
        "   set newname for datafile  4 to ",
        " \"+DATAS\";",
        "   backup as copy reuse",
        "   datafile  1 auxiliary format ",
        " \"+DATAS\"   datafile ",
        " 2 auxiliary format ",
        " \"+DATAS\"   datafile ",
        " 3 auxiliary format ",
        " \"+DATAS\"   datafile ",
        " 4 auxiliary format ",
        " \"+DATAS\"   ;",
        "   sql 'alter system archive log current';",
        "}",
        "executing Memory Script",
        "",
        "executing command: SET NEWNAME",
        "",
        "renamed tempfile 1 to +DATAS in control file",
        "",
        "executing command: SET NEWNAME",
        "",
        "executing command: SET NEWNAME",
        "",
        "executing command: SET NEWNAME",
        "",
        "executing command: SET NEWNAME",
        "",
        "Starting backup at 04-APR-25",
        "channel prmy1: starting datafile copy",
        "input datafile file number=00002 name=+DATA/ORCL/DATAFILE/sysaux.262.1196268195",
        "channel prmy2: starting datafile copy",
        "input datafile file number=00001 name=+DATA/ORCL/DATAFILE/system.261.1196268193",
        "channel prmy3: starting datafile copy",
        "input datafile file number=00003 name=+DATA/ORCL/DATAFILE/undotbs1.263.1196268197",
        "channel prmy4: starting datafile copy",
        "input datafile file number=00004 name=+DATA/ORCL/DATAFILE/users.265.1196268203",
        "output file name=+DATAS/ORCLS/DATAFILE/users.334.1197553613 tag=TAG20250404T134653",
        "channel prmy4: datafile copy complete, elapsed time: 00:00:01",
        "output file name=+DATAS/ORCLS/DATAFILE/sysaux.277.1197553613 tag=TAG20250404T134653",
        "channel prmy1: datafile copy complete, elapsed time: 00:00:04",
        "output file name=+DATAS/ORCLS/DATAFILE/system.269.1197553613 tag=TAG20250404T134653",
        "channel prmy2: datafile copy complete, elapsed time: 00:00:04",
        "output file name=+DATAS/ORCLS/DATAFILE/undotbs1.335.1197553613 tag=TAG20250404T134653",
        "channel prmy3: datafile copy complete, elapsed time: 00:00:04",
        "Finished backup at 04-APR-25",
        "",
        "sql statement: alter system archive log current",
        "",
        "contents of Memory Script:",
        "{",
        "   switch clone datafile all;",
        "}",
        "executing Memory Script",
        "",
        "datafile 1 switched to datafile copy",
        "input datafile copy RECID=1 STAMP=1197553617 file name=+DATAS/ORCLS/DATAFILE/system.269.1197553613",
        "datafile 2 switched to datafile copy",
        "input datafile copy RECID=2 STAMP=1197553617 file name=+DATAS/ORCLS/DATAFILE/sysaux.277.1197553613",
        "datafile 3 switched to datafile copy",
        "input datafile copy RECID=3 STAMP=1197553617 file name=+DATAS/ORCLS/DATAFILE/undotbs1.335.1197553613",
        "datafile 4 switched to datafile copy",
        "input datafile copy RECID=4 STAMP=1197553617 file name=+DATAS/ORCLS/DATAFILE/users.334.1197553613",
        "Finished Duplicate Db at 04-APR-25",
        "released channel: prmy1",
        "released channel: prmy2",
        "released channel: prmy3",
        "released channel: prmy4",
        "released channel: stby",
        "",
        "Recovery Manager complete.",
        "RMAN completed successfully.",
        "Dataguard setup completed successfully!"
    ]
}

TASK [create_standby : Checking if Dataguard restore was already done]
********************************************************************************************************************
skipping: [primary-db]
ok: [standby-db]

TASK [create_standby : Set fact for primary if restore was already done on standby]
********************************************************************************************************
skipping: [primary-db]
ok: [standby-db]

TASK [create_standby : Fetch post restore status from standby to primary]
*****************************************************************************************************************
skipping: [standby-db]
ok: [primary-db]

TASK [create_standby : Checking if Dataguard post restore was already done]
***************************************************************************************************************
skipping: [primary-db]
ok: [standby-db]

TASK [create_standby : Copy RMAN Restore Post-Processing Script]
**************************************************************************************************************************
skipping: [primary-db]
changed: [standby-db]

TASK [create_standby : Run RMAN Restore Post-Processing Script]
***************************************************************************************************************************
skipping: [primary-db]
ok: [standby-db]

TASK [create_standby : Display RMAN Restore Post-Processing Script Output]
****************************************************************************************************************
skipping: [primary-db]
ok: [standby-db] => {
    "msg": [
        "RMAN Listener stop successful",
        "RMAN Listener successfully stopped",
        "RMAN Listener directory removed successfully",
        "ORA-01109: database not open",
        "",
        "",
        "Database dismounted.",
        "ORACLE instance shut down.",
        "ORACLE instance started.",
        "",
        "Total System Global Area 1.0335E+10 bytes",
        "Fixed Size\t\t   13842592 bytes",
        "Variable Size\t\t 1610612736 bytes",
        "Database Buffers\t 8690597888 bytes",
        "Redo Buffers\t\t   19660800 bytes",
        "Database mounted.",
        "",
        "System altered.",
        "",
        "",
        "System altered.",
        "",
        "",
        "System altered.",
        "",
        "Dataguard recovery process started successfully"
    ]
}

TASK [create_standby : Checking if Dataguard post restore was already done on standby]
*****************************************************************************************************
skipping: [primary-db]
ok: [standby-db]

TASK [create_standby : Set fact for post restore status]
**********************************************************************************************************************************
skipping: [primary-db]
ok: [standby-db]

TASK [create_standby : Retrieve post restore status from standby]
*************************************************************************************************************************
skipping: [standby-db]
ok: [primary-db]

TASK [create_standby : Set fact for condition evaluation]
*********************************************************************************************************************************
ok: [primary-db]
ok: [standby-db]

TASK [create_standby : Copy Protection Mode Post-Processing Script on primary]
*************************************************************************************************************
skipping: [standby-db]
changed: [primary-db]

TASK [create_standby : Execute Protection Mode Post-Processing Script on primary]
**********************************************************************************************************
skipping: [standby-db]
ok: [primary-db]

TASK [create_standby : Display Protection Mode Post-Processing Output]
********************************************************************************************************************
skipping: [standby-db]
ok: [primary-db] => {
    "msg": [
        "PROTECTION_MODE:MAXIMUM PERFORMANCE",
        "Dataguard successfully configured with PROTECTION_MODE: MAXIMUM PERFORMANCE",
        "Dataguard configured successfully"
    ]
}
```

```
TASK [post_sync_check : Run initialization tasks]
********************************************************************************************************************************************
skipping: [primary-db]
skipping: [standby-db]

TASK [post_sync_check : Copy Data Guard sync check script to primary]
************************************************************************************************************************
skipping: [standby-db]
changed: [primary-db]

TASK [post_sync_check : Copy Data Guard sync check script to standby]
************************************************************************************************************************
skipping: [primary-db]
changed: [standby-db]

TASK [post_sync_check : Execute Data Guard sync check on Primary]
****************************************************************************************************************************
skipping: [standby-db]
ok: [primary-db]

TASK [post_sync_check : Execute Data Guard sync check on Standby]
****************************************************************************************************************************
skipping: [primary-db]
ok: [standby-db]

TASK [post_sync_check : Fetch Standby Sequence on Primary]
***********************************************************************************************************************************
ok: [primary-db]
skipping: [standby-db]

TASK [post_sync_check : Compare sequence numbers between primary and standby]
****************************************************************************************************************
ok: [primary-db]

TASK [post_sync_check : Display message if standby is in sync]
*******************************************************************************************************************************
skipping: [primary-db]
skipping: [standby-db]

TASK [post_sync_check : End play if standby is in sync with primary]
*************************************************************************************************************************
skipping: [primary-db]

TASK [post_sync_check : Continue if standby is not in sync]
**********************************************************************************************************************************
ok: [primary-db] => {
    "msg": "Standby is not in sync with Primary. Proceeding with further actions."
}
ok: [standby-db] => {
    "msg": "Standby is not in sync with Primary. Proceeding with further actions."
}

TASK [post_sync_check : Copy Data Guard sync script to primary and standby]
******************************************************************************************************************
changed: [standby-db]
changed: [primary-db]

TASK [post_sync_check : Execute Data Guard sync check on Primary]
****************************************************************************************************************************
skipping: [standby-db]
ok: [primary-db]

TASK [post_sync_check : Set fact for Primary output]
*****************************************************************************************************************************************
ok: [primary-db]
skipping: [standby-db]

TASK [post_sync_check : Fail the play if Data Guard is not in sync]
**************************************************************************************************************************
skipping: [primary-db]

TASK [post_sync_check : End the play if Data Guard is not in sync]
***************************************************************************************************************************
skipping: [primary-db]

TASK [post_sync_check : Extract Primary Sequence Number]
*************************************************************************************************************************************
skipping: [standby-db]
ok: [primary-db]

TASK [post_sync_check : Debug Primary Sequence Number]
***************************************************************************************************************************************
skipping: [standby-db]
ok: [primary-db] => {
    "msg": "PRIMARY_SEQ:88\nSTATUS:VALID,ERROR:\nPrimary database sync status is valid"
}

TASK [post_sync_check : Execute Data Guard sync check on Standby]
****************************************************************************************************************************
skipping: [primary-db]
ok: [standby-db]

TASK [post_sync_check : Extract Standby Sequence Number]
*************************************************************************************************************************************
skipping: [primary-db]
ok: [standby-db]

TASK [post_sync_check : Debug Standby Sequence Number]
***************************************************************************************************************************************
skipping: [primary-db]
ok: [standby-db] => {
    "msg": "MRP_RUNNING:0\nMRP0 process not running. Enabling managed recovery...\nMRP_RUNNING:1\nWaiting for 180 seconds to allow standby to
sync...\nSTANDBY_SEQ:88"
}

TASK [post_sync_check : Fetch Standby Sequence on Primary]
***********************************************************************************************************************************
ok: [primary-db]
skipping: [standby-db]

TASK [post_sync_check : Debug Standby and primary Sequence Number]
***************************************************************************************************************************
ok: [primary-db] => {
    "msg": "Primary:  ['88'], Standby: ['88']"
}
skipping: [standby-db]

TASK [post_sync_check : Fail if Data Guard is out of sync]
***********************************************************************************************************************************
skipping: [primary-db]
```

```
TASK [post_sync_check : Query Data Guard Sync Status on Standby]
****************************************************************************************************************************
skipping: [primary-db]
ok: [standby-db]

TASK [post_sync_check : Debug Raw SQL Output]
***********************************************************************************************************************************************
skipping: [primary-db]
ok: [standby-db] => {
    "dg_sync_status.stdout_lines": [
        "",
        "    Thread Last Sequence Received Last Sequence Applied Difference",
        "---------- ---------------------- --------------------- ----------",
        "\t 1\t\t       87\t\t     87 \t 0"
    ]
}

TASK [post_sync_check : Extract Data Guard Sync Status]
*************************************************************************************************************************************
skipping: [primary-db]
ok: [standby-db]

TASK [post_sync_check : Print Data Guard Sync Status]
***************************************************************************************************************************************
skipping: [primary-db]
ok: [standby-db] => {
    "msg": "**Data Guard Sync Check**\n-------------------------------------\n**Thread:** 1\n**Last Sequence Received:** 87\n**Last Sequence
Applied:** 87\n**Difference:** 0\n------------------------------------\n**Status:** In Sync\n"
}

PLAY RECAP
*************************************************************************************************************************************************
*********************************
primary-db                 : ok=57   changed=9    unreachable=0    failed=0    skipped=65   rescued=0    ignored=0
standby-db                 : ok=59   changed=8    unreachable=0    failed=0    skipped=55   rescued=0    ignored=0
```

12. Executing Oracle collection using Ansible Automation Platform 2 (AAP2)

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

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

At first using “ansible-builder” build the Container Image .For more info regarding ansible-builder refer to below

https://access.redhat.com/documentation/en-us/red_hat_ansible_automation_platform/2.0-ea/html-single/ansible_builder_guide/index

To execute this playbook from AAP2 follow below steps. An example document is provided here:
Github: https://github.com/nava-dba/ansible-power-aix-oracle-dba

12.1 Build the podman environment

```
su – awx
Create directory
mkdir oracle_rac_dataguard_ee
```

vi execution-environment.yml

```
# Execution Environment
---
version: 3

images:
  base_image:
    name: registry.redhat.io/ansible-automation-platform-24/ee-minimal-rhel8:latest
options:
  package_manager_path: /usr/bin/microdnf
additional_build_steps:
  append_base:
   - RUN microdnf install gcc python39-devel libnsl* libaio* find* which* sudo dnf
dependencies:
 galaxy: requirements.yml
```

vi requirements.yml

```
---
collections:
  - ibm.power_aix
  - ansible.utils
```

After creating execution-environment.yml file use ansible-builder to build container image run below

```
$ ansible-builder build -t oracle_rac_dataguard_ee -f execution-environment.yml
Running command:
  podman build -f context/Containerfile -t oracle_rac_dataguard_ee context
Complete! The build context can be found at: /var/lib/awx/oracle_rac_dataguard_ee/context
```

Validate the image:

```
$ podman images
REPOSITORY                                                         TAG        IMAGE ID      CREATED             SIZE
localhost/oracle_rac_dataguard_ee                                    latest       8d597e0a1bab  About a minute ago  493 MB
localhost/oracle_aix_ee                                            latest     a7d4b0992ea4  2 months ago        475 MB
localhost/powerodba                                                latest     3d1b75b3ee16  3 months ago        910 MB
registry.redhat.io/ansible-automation-platform-24/ee-supported-rhel8  latest     b2d26de2d8de  11 months ago       1.79 GB
registry.redhat.io/ansible-automation-platform-24/ee-minimal-rhel8    latest     c239714e9480  11 months ago       380 MB
quay.io/ansible/ansible-runner                                     latest     bec0dc171168  2 years ago         816 MB
```

12.2 Execute Playbook via CLI - Ansible-Navigator

We can use ansible-navigator for executing the playbook in CLI using execution environments (Container image).

Go to power_aix_oracle collection and create ansible-navigator.yaml file

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

$ ansible-navigator run dataguard_playbook.yml --pp=missing -m stdout
```

12.3 Executing the Playbook from Ansible Controller AAP2 using execution environment via GUI

13. Conclusion

This Ansible playbook automates Oracle Single Instance Data Guard setup, ensuring a seamless and repeatable deployment process. The setup enhances database resilience with minimal manual intervention.