

Table of Contents

1.	Introduction	2
2.	Versions.....	2
3.	Architecture Diagram	2
4.	Prerequisites.....	2
5.	Ansible Playbook Setup and Execution Steps	3
5.1	Install Ansible Controller on your preferred operation system.	3
5.2	Setup ssh Equivalence between the Ansible controller and managed host (Primary and Standby) server	3
5.3	Download playbook from Galaxy.....	3
6.	Ansible Playbook Description	3
6.1	Playbook Files	3
6.2	Ansible Playbook Structure.....	4
6.3	Ansible Playbook Roles.....	4
7.	Limitations	5
8.	Methods for Configuring Standby Database	5
8.1	RMAN ACTIVE Duplicate	5
8.2	RMAN Backup-Based	5
8.3	Parameters for each methods.....	5
9.	Preparing to run the Dataguard playbook	5
9.1	Change directory to the galaxy_collection directory	5
9.2	Update the inventory	5
9.3	Update the variable file	5
9.4	Update the vaults.yml file with sys password and encrypt it with ansible vault.....	5
10.	Execute the playbook.....	5
10.1	RMAN ACTIVE DUPLICATE	5
10.2	RMAN BACKUP BASED	6
11.	Executing Oracle collection using Ansible Automation Platform 2 (AAP2).....	7
11.1	Build the podman environment	7
11.2	Execute Playbook via CLI - Ansible-Navigator	7
11.3	Executing the Playbook from Ansible Controller AAP2 using execution environment via GUI	7
12.	Troubleshooting	25
13.	Conclusion	26

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 has 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.

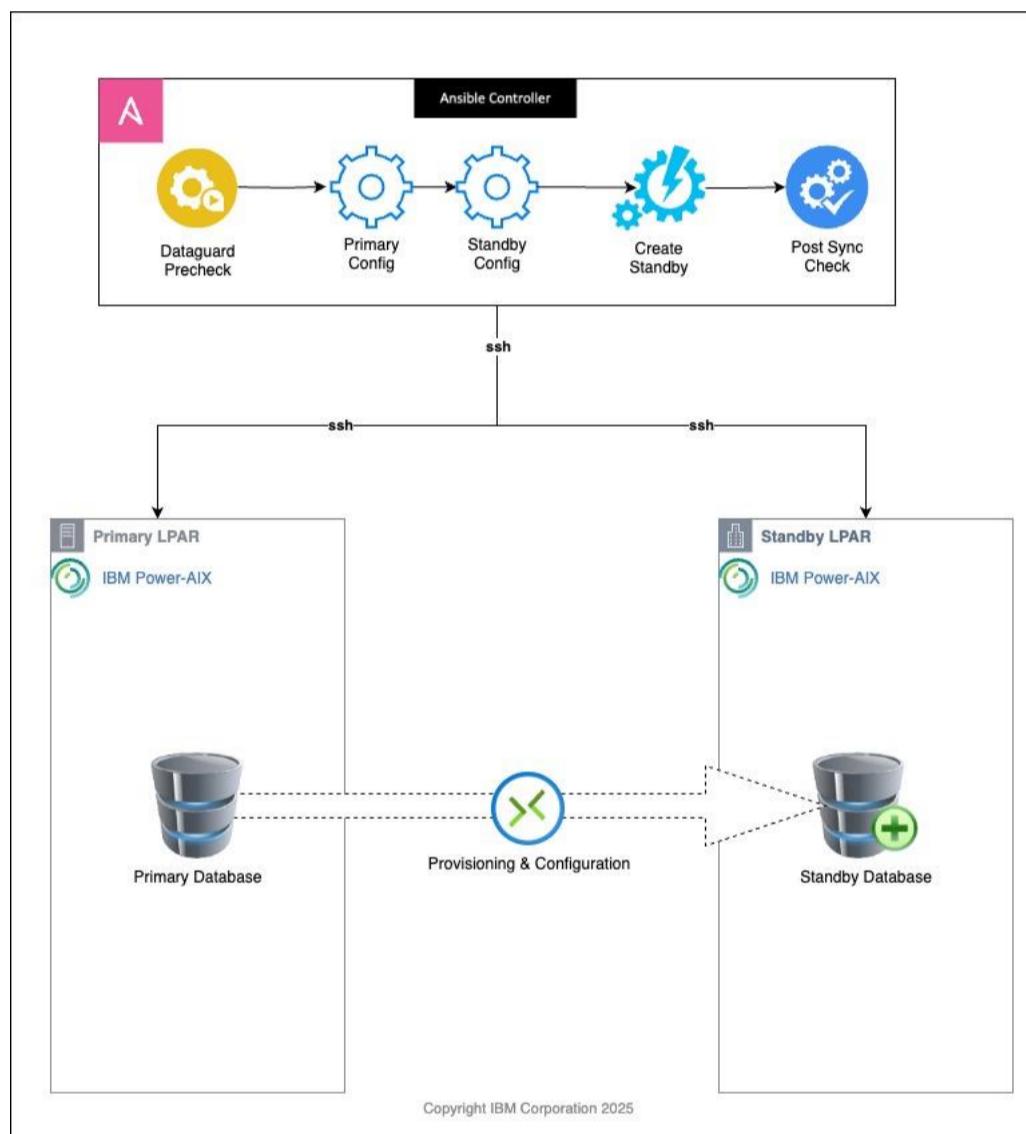


Fig: 1 Oracle Single Instance Dataguard Implementation using Ansible PowerODBA Collection

4. Prerequisites

This playbook assumes the following:

- Oracle 19c Software installed on both nodes primary and standby , please visit https://galaxy.ansible.com/ui/repo/published/ibm/power_aix_oracle/ for installing Oracle Single Instance database 19c on new AIX operating system and create test database on AIX filesystem JFS2 or Oracle ASM.
- Ansible installed on the control node
- Passwordless SSH setup between control node and both managed hosts AIX database servers
- Requires network connectivity between the primary and standby databases servers , hostnames provided in variable file should be pingable from each other
- Ensure sufficient storage, memory and CPU resources are available on both sites primary and standby servers
- ASM disk groups or JFS file systems needed for standby database should be configured
- Primary database should be in ARCHIVE_LOG mode
- REDO log sizes should be identical for primary database
- Patch levels between primary and standby for Oracle Clusterware and database should be same , to ignore please use ignore_precheck = true
- On the standby server with ASM, set up passwordless SSH (SSH equivalence) from the Oracle user to the Grid user in case of role separation - different user for grid stack and oracle DB home stack
 ssh grid_user@standby_hostname should be enabled from oracle_user in standby server
- Update the backup_location in vars if different in both primary and standby before invoking the respective play
- Make sure oracle database software owner has appropriate permission to \$ORACLE_BASE/admin directory on standby standby server.
Preferably :
 chmod -R 775 \$ORACLE_BASE/admin
 chown -R <oracle_user>:<oracle_user_group> \$ORACLE_BASE/admin

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 between the Ansible controller and managed host (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_dba/

Download the `power_aix_oracle_dba` 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_dba`

The above command will install `power_aix_oracle_dba` 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 - “`ansible-collection-install-dir`”/`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 Dataguard setup 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. **dataguard_primary_config:**
 - Configure Dataguard parameters in primary database
 - Enables FORCE LOGGING
 - Creates the Standby Redo Logs
3. **dataguard_standby_config:**
 - Configure Dataguard parameters in standby database
4. **dataguard_create_standby:**
 - Creates standby using RMAN ACTIVE DUPLICATE or RMAN Backup
 - Starts the standby database in managed recovery mode
5. **Dataguard_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 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	
dataguard_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	
dataguard_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 ACTIVE 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	
dataguard_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 ACTIVE duplicate method only	
dataguard_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 non-multitenant database running on AIX
- If GAP between primary and standby 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/ standby_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 both primary and standby
- In this release, Data Guard operates in MAXIMUM PERFORMANCE mode only and does not support other modes
- Multiple standby database is not supported in this release
- Use nohup to avoid timeouts while execution of playbook
- These playbooks will create two directories inside 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 ACTIVE Duplicate

1. Automation leverages 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 (dataguard_rman_backup_playbook.yml) or manually
2. Transfer backup files to the standby server manually and update the backup_location in the variable file.
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-dba
```

9.2 Update the inventory

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

[standby]
standby-db ansible_host=222.22.22.222
```

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 , this ansible play will create a physical standby database using RMAN DUPLICATE FROM ACTIVE DATABASE command without shutting down the primary and using primary active database files.

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: dataguard_primary_config
      tags: dataguard_primary_config

    - role: dataguard_standby_config
      tags: dataguard_standby_config

    - role: dataguard_create_standby
      tags: dataguard_create_standby

    - role: dataguard_post_sync_check
      tags: dataguard_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,dataguard_primary_config,
dataguard_standby_config,dataguard_create_standby,dataguard_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
dataguard_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 dataguard_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,dataguard_primary_config,
dataguard_standby_config,dataguard_create_standby,dataguard_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
dataguard_post_sync_check

```

Sample Execution logs: Please find the sample execution logs in - docs/ dataguard_sample_execution_logs/ directory

11. 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:

11.1 Build the podman environment

```
su - awx
Create directory
mkdir oracle_si_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
```

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

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

Validate the image:

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
localhost/oracle_si_dataguard_ee	latest	35c6e5dae65f	47 seconds ago	362 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

11.2 Execute Playbook via CLI - Ansible-Navigator

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

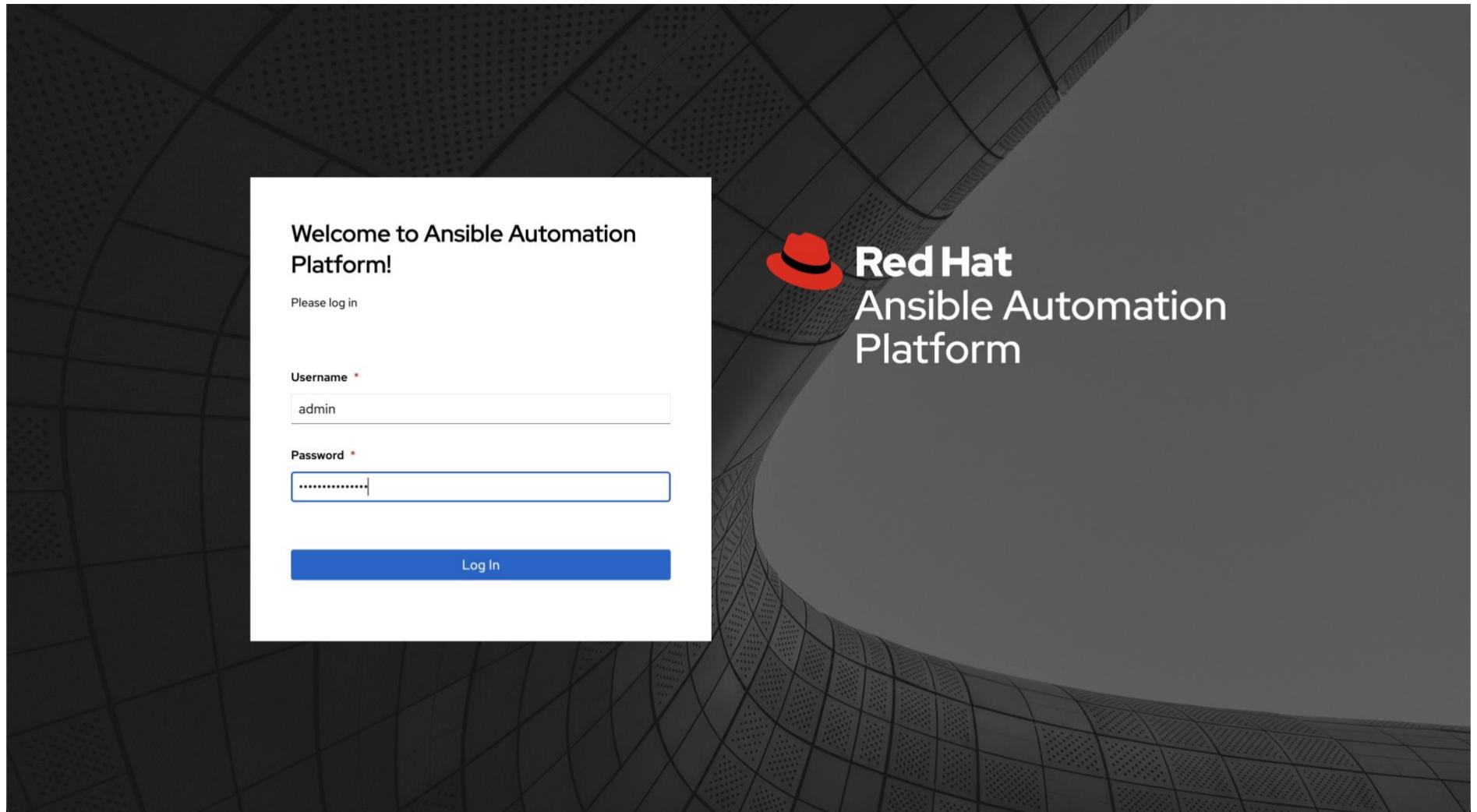
Go to power_aix_oracle collection and create ansible-navigator.yaml file

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

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

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

1. Login to the Ansible controller, provide the username and password.



2. To create a new execution environment, please click on the "Execution Environments" option under the Administration dropdown and click on Add

A screenshot of the platform's interface. The left sidebar has sections like Resources (Templates, Credentials, Projects, Inventories, Hosts), Access (Organizations, Users, Teams), Administration (Credential Types, Notifications, Management Jobs, Instance Groups, Instances, Applications, Execution Environments, Topology View, Settings). The "Execution Environments" section is highlighted. The main content area shows a table titled "Execution Environments" with columns: Name, Image, Organization, and Actions. Five rows are listed:

Name	Image	Organization	Actions
Control Plane Execution Environment	registry.redhat.io/ansible-automation-platform-24/ee-supported-rhel8:latest	Globally Available	
Default execution environment	registry.redhat.io/ansible-automation-platform-24/ee-supported-rhel8:latest	Globally Available	
Minimal execution environment	registry.redhat.io/ansible-automation-platform-24/ee-minimal-rhel8:latest	Globally Available	
PODBA-EE	localhost/podba-ee	Default	
PODBA RAC Upgrade EE	localhost/oracle_rac_upgrade_ee	Default	

Please provide the following details to create a new execution environment:
Name: [Enter the name of the execution environment]
Image: [Specify the Docker image for the execution environment]
Pull Details: [Provide any additional details for pulling the image]
Once provided, click on "Save" to create the new execution environment.

The screenshot shows the Red Hat Ansible Automation Platform interface. The left sidebar is dark-themed and includes sections for Resources (Templates, Credentials, Projects, Inventories, Hosts), Access (Organizations, Users, Teams), Administration (Credential Types, Notifications, Management Jobs, Instance Groups, Instances, Applications, Execution Environments, Topology View), and Settings. The 'Execution Environments' section is currently selected under Administration. The main content area is titled 'Create new execution environment'. It contains fields for 'Name' (with a required asterisk), 'Image' (with a required asterisk), 'Pull' (set to '-----'), 'Description', 'Organization' (set to 'Default'), and 'Registry credential'. A note below the organization field says 'Leave this field blank to make the execution environment globally available.' At the bottom are 'Save' and 'Cancel' buttons.

Once saved, we can see the following details to crosscheck.

This screenshot shows the same interface after the execution environment has been created. The 'Name' field now contains 'PODBA Dataguard EE', the 'Image' field contains 'localhost/oracle_si_dataguard_ee', and the 'Pull' dropdown is set to 'Only pull the image if not present before running.'. The other fields remain the same. The 'Save' and 'Cancel' buttons are at the bottom.

Once saved, the container image will be listed in Execution Environments

- To create Projects, click on the "Projects" option under the Resources dropdown and then click on the "Add" button to create Projects.

Please provide the following details:
Name: [Provide the name of your project]
Description: [Briefly describe your project]
Organization: [Select the organization for your project]
Execution Environment: [Select the corresponding execution environment created earlier]
Source Control Type: Git
Source Control URL: https://github.com/nav-a-dba/Upgrade_rac_DB [where the code is pushed]

Once provided the details, please click on “Save”

Red Hat Ansible Automation Platform

admin

Resources

- Templates
- Credentials
- Projects**
- Inventories
- Hosts

Access

- Organizations
- Users
- Teams

Administration

- Credential Types
- Notifications
- Management Jobs
- Instance Groups
- Instances
- Applications
- Execution Environments
- Topology View

Settings

Projects

Create New Project

Name *	Description	Organization *
<input type="text"/>	<input type="text"/>	<input type="text"/> Default
Execution Environment ⓘ	Source Control Type *	Content Signature Validation Credential ⓘ
<input type="text"/>	<input type="button" value="Choose a Source Control Type"/>	<input type="text"/>
<input type="button" value="Save"/> <input type="button" value="Cancel"/>		

Red Hat Ansible Automation Platform

admin

Schedules

Activity Stream

Workflow Approvals

Host Metrics

Resources

- Templates
- Credentials
- Projects**
- Inventories
- Hosts

Access

- Organizations
- Users
- Teams

Administration

- Credential Types
- Notifications
- Management Jobs
- Instance Groups
- Instances
- Applications
- Execution Environments
- Topology View

Projects > DG Setup

Edit Details

Name *	Description	Organization *
DG Setup	<input type="text"/>	<input type="text"/> Default
Execution Environment ⓘ	Source Control Type *	Content Signature Validation Credential ⓘ
<input type="text"/> PODBA Dataguard EE	<input type="text"/> Git	<input type="text"/>
Type Details		
Source Control URL * ⓘ	Source Control Branch/Tag/Commit ⓘ	Source Control Refspec ⓘ
<input type="text"/> https://github.com/nav-a-dba/dataguard_setup	<input type="text"/>	<input type="text"/>
Source Control Credential		
<input type="text"/>		
Options		
<input checked="" type="checkbox"/> Clean ⓘ <input type="checkbox"/> Delete ⓘ <input type="checkbox"/> Track submodules ⓘ <input checked="" type="checkbox"/> Update Revision on Launch ⓘ <input type="checkbox"/> Allow Branch Override ⓘ		
Option Details		
Cache Timeout ⓘ		
<input type="text"/> 0		
<input type="button" value="Save"/> <input type="button" value="Cancel"/>		

Once saved, we can see the following details to crosscheck.

Details

Last Job Status	Successful	Name	DG Setup	Organization	Default
Source Control Type	Git	Source Control Revision	29cedfa	Source Control URL	https://github.com/navanya-dba/dataguard_setup
Cache Timeout	0 Seconds	Default Execution Environment	PODBA RAC Upgrade EE	Project Base Path	/var/lib/awx/projects
Playbook Directory	_161_dg_setup	Created	19/03/2025, 01:43:59 by admin	Last Modified	06/04/2025, 14:48:45 by admin
Enabled Options	Discard local changes before syncing Update revision on job launch				

- Git pull should be success [Last Job Status]

4. To create Inventories, click on the "Inventories" option under the Resources dropdown and then click on the "Add" button to create Inventories.

Inventories

Name	Sync Status	Type	Organization	Actions
PODBA inventory	Disabled	Inventory	Default	
PODBA Upgrade inventory	Disabled	Inventory	Default	

Please provide the following details:

Name: [Enter the name of your project]

Description: [Enter a brief description of your project]

Organization: [Select the organization for your project]

After providing the required information, click on "Save".

Inventories

Create new inventory

Name *	Description	Organization *
<input type="text"/>	<input type="text"/>	<input type="text"/> Default
Instance Groups		
<input type="text"/>		
Labels <small>①</small>		
<input type="text"/>		
Options		
<input type="checkbox"/> Prevent Instance Group Fallback <small>②</small>		
Variables <small>③</small> <input type="button" value="YAML"/> <input type="button" value="JSON"/>		
<pre>1 ---</pre>		
<input type="button" value="Save"/> <input type="button" value="Cancel"/>		

Once saved, we can see the following details to crosscheck.

Inventories > DG Setup

Details

Back to Inventories		Details	Access	Groups	Hosts	Sources	Jobs	Job Templates		
Name	DG Setup	Description	Primary and standby hosts							
Organization	Default	Type	Inventory							
Variables <small>①</small>	<input type="button" value="YAML"/> <input type="button" value="JSON"/>	Total hosts	2							
		1 ---								
Created	27/02/2025, 10:13:18 by admin	Last Modified	27/02/2025, 10:13:18 by admin							
		<input type="button" value="Edit"/>	<input type="button" value="Delete"/>							

5. Create Two Hosts to support RAC for two nodes database

To create First hosts, click on the “Hosts” Option Under the resources dropdown and then click on the “Add” button to Create First Hosts

The screenshot shows the 'Hosts' list page in the Red Hat Ansible Automation Platform. The left sidebar has 'Hosts' selected under 'Resources'. The main area displays three hosts: 'orahostb' (Description: 'PODBA inventory'), 'p227n242.pbm.ihost.com' (Description: 'PODBA Upgrade inventory'), and 'p227n243.pbm.ihost.com' (Description: 'PODBA Upgrade inventory'). Each host has an 'On' toggle switch and an edit icon.

Please provide the following information for the host 1:

Name: [Enter the name of the hostname of target machine, not with any name]

Description: [Enter a brief description of your project]

Inventory: [Select the inventory that was earlier created]

Variables: ansible-host: <IP Address>

After providing the required information, click on "Save".

The screenshot shows the 'Create New Host' dialog. The left sidebar has 'Hosts' selected under 'Resources'. The dialog form includes fields for 'Name' (with a red asterisk), 'Description', and 'Inventory' (with a search icon). Below these are 'Variables' sections for 'YAML' and 'JSON'. A preview area shows two variables: '1 ---' and '2'. At the bottom are 'Save' and 'Cancel' buttons.

Once saved, we can see the following details to crosscheck.

Schedules
Activity Stream
Workflow Approvals
Host Metrics
Resources
Templates
Credentials
Projects
Inventories
Hosts
Access
Organizations
Users
Teams
Administration
Credential Types
Notifications
Management Jobs
Instance Groups
Instances
Applications
Execution Environments
Topology View

Hosts > primary-db
Details

Back to Hosts Details Facts Groups Jobs

On

Name: primary-db Activity: ✓✓✓✓✗ Description: primary-db

Inventory: DG Setup Created: 27/02/2025, 10:14:20 by admin Last Modified: 27/02/2025, 10:14:20 by admin

Variables: YAML JSON

1 ---
2 ansible_host: 129.40.76.243

Edit Delete

6. To create Second hosts, click on the “Hosts” Option Under the resources dropdown and then click on the “Add” button to Create Second Hosts

Views
Dashboard
Jobs
Schedules
Activity Stream
Workflow Approvals
Host Metrics
Resources
Templates
Credentials
Projects
Inventories
Hosts
Access
Organizations
Users
Teams
Administration
Credential Types
Notifications

Hosts

Name Description Inventory Actions

Name	Description	Inventory	Actions
orahostb		PODBA inventory	On Edit
p227n242.pbm.ihost.com	rac21	PODBA Upgrade inventory	On Edit
p227n243.pbm.ihost.com	rac22	PODBA Upgrade inventory	On Edit

Please provide the following information for the host 2:
Name: [Enter the name of the hostname of second target machine, not with any name]
Description: [Enter a brief description of your project]
Inventory: [Select the inventory that was earlier created]
Variables: ansible-host
After providing the required information, click on "Save".

The screenshot shows the 'Create New Host' form in the Red Hat Ansible Automation Platform. The left sidebar has sections for Views (Dashboard, Jobs, Schedules, Activity Stream, Workflow Approvals, Host Metrics), Resources (Templates, Credentials, Projects, Inventories, Hosts), Access (Organizations, Users, Teams), and Administration (Credential Types, Notifications). The 'Hosts' tab under Resources is selected. The main area shows a 'Create New Host' form with fields for Name (required), Description, and Inventory (with a search icon). Below these are tabs for Variables (YAML, JSON) and a code editor containing host variables. At the bottom are 'Save' and 'Cancel' buttons.

Once saved, we can see the following details to crosscheck.

The screenshot shows the 'Details' page for the host 'standby-db'. The left sidebar is identical to the previous screenshot. The main area shows the host's status (On), name (standby-db), activity (green), inventory (DG Setup), description (standby-db), creation date (27/02/2025, 10:14:37 by admin), and last modified date (27/02/2025, 10:14:37 by admin). It also shows host variables (YAML, JSON) and a code editor with the variable 'ansible_host: 129.40.76.242'. At the bottom are 'Edit' and 'Delete' buttons.

7. Now, to create Credentials, click on the "Credentials" option under the Resources dropdown and then click on the "Add" button to create Credentials

Name	Type	Actions
Ansible Galaxy	Ansible Galaxy/Automation Hub API Token	
oracle-cred	Machine	
oracle_rac_upgrade_cred	Machine	

Please provide the following information:

Name: [Enter the name of your credential]

Description: [Enter a brief description of your credential]

Organization: [Select the organization for your credential]

Credential Type: Machine

Username: [Enter the username for the machine]

Password: [Enter the password for the machine]

After providing the required information, click on "Save".

Once saved, we can see the following details to crosscheck.

- Now, to create Templates, click on the "Templates" option under the Resources dropdown and then click on the "Add" button to create Templates.

Please provide the following details if you want all the roles to be executed:

Name: [Enter the name of your job]

Description: [Enter a brief description of your job]

Job Type: Run

Inventory: [Select the inventory that was earlier created]

Project: [Select the project that was earlier created]

Execution Environment: [Select the execution environment that was earlier created]

Playbook: [Enter the name of your playbook file]

Variables: ansible_ssh_user: oracle

After providing the required information, click on "Save".

Red Hat Ansible Automation Platform

Views

- Dashboard
- Jobs
- Schedules
- Activity Stream
- Workflow Approvals
- Host Metrics

Resources

- Templates**
- Credentials
- Projects
- Inventories
- Hosts

Access

- Organizations
- Users
- Teams

Administration

- Credential Types
- Notifications

Templates

Create New Job Template

Name *	Description	Job Type * ⓘ	<input type="checkbox"/> Prompt on launch
Inventory * ⓘ	Project * ⓘ	Execution Environment ⓘ	<input type="checkbox"/> Prompt on launch
Playbook * ⓘ		<input type="checkbox"/> Prompt on launch	
Credentials ⓘ		<input type="checkbox"/> Prompt on launch	
Labels ⓘ		<input type="checkbox"/> Prompt on launch	
Variables ⓘ		<input type="checkbox"/> Prompt on launch	<input type="checkbox"/> X
YAML JSON			
1 ---	2 ---		
Forks ⓘ	<input type="checkbox"/> Prompt on launch	Limit ⓘ	<input type="checkbox"/> Prompt on launch
		<input type="checkbox"/> Prompt on launch	Verbosity ⓘ
			<input type="checkbox"/> Prompt on launch

Once saved, we can see the following details to crosscheck.

Red Hat Ansible Automation Platform

Activity Stream

Workflow Approvals

Host Metrics

Resources

- Templates**
- Credentials
- Projects
- Inventories
- Hosts

Access

- Organizations
- Users
- Teams

Administration

- Credential Types
- Notifications
- Management Jobs
- Instance Groups
- Instances
- Applications
- Execution Environments
- Topology View

Templates > DG RMAN Primary DB backup

Details

Back to Templates		Details	Access	Notifications	Schedules	Jobs	Survey
Name	DG RMAN Primary DB backup	Description	Backup Primary DB for DG Setup	Job Type ⓘ	run		
Organization	Default	Inventory ⓘ	primary-db	Project ⓘ	DG Setup		
Execution Environment ⓘ	PODBA Dataguard EE	Playbook ⓘ	rman_backup_playbook.yml	Forks ⓘ	0		
Verbosity ⓘ	0 (Normal)	Timeout ⓘ	0	Show Changes ⓘ	Off		
Job Slicing ⓘ	1	Created	27/02/2025, 10:10:30 by admin	Last Modified	09/04/2025, 10:38:03 by admin		
Credentials ⓘ	SSH: DG Setup						
Job Tags ⓘ	backup_primary_db						
Variables ⓘ	<input type="checkbox"/> YAML <input type="checkbox"/> JSON	<pre> 1##### 2 # This File contain all the Global Variables required to do the dataguard configuration for 19c database. 3 # Please update the variables by reading the comments provided at the 4 # beginning of each section. </pre>					
		<input type="button" value="Edit"/>	<input type="button" value="Launch"/>	<input type="button" value="Delete"/>			

Red Hat Ansible Automation Platform

Activity Stream Workflow Approvals Host Metrics

Resources **Templates**

Credentials Projects Inventories Hosts

Access Organizations Users Teams

Administration Credential Types Notifications Management Jobs Instance Groups Instances Applications Execution Environments Topology View Settings

Templates > DG precheck Details

Back to Templates Details Access Notifications Schedules Jobs Survey

Name	DG precheck	Description	precheck Primary & standby DB for DG Setup	Job Type ⓘ	run
Organization	Default	Inventory ⓘ	DG Setup	Project ⓘ	DG Setup
Execution Environment ⓘ	PODBA Dataguard EE	Playbook ⓘ	dataguard_playbook.yml	Forks ⓘ	0
Verbosity ⓘ	0 (Normal)	Timeout ⓘ	0	Show Changes ⓘ	Off
Job Slicing ⓘ	1	Created	27/02/2025, 13:15:51 by admin	Last Modified	09/04/2025, 10:39:38 by admin
Credentials ⓘ	SSH: DG Setup				
Job Tags ⓘ	dataguard_precheck				
Variables ⓘ	YAML JSON	<pre>1- ##### 2 # This File contain all the Global Variables required to do the dataguard configuration for 19c database. 3 # Please update the variables by reading the comments provided at the 4 # beginning of each section.</pre>			
Edit Launch Delete					

Red Hat Ansible Automation Platform

Activity Stream Workflow Approvals Host Metrics

Resources **Templates**

Credentials Projects Inventories Hosts

Access Organizations Users Teams

Administration Credential Types Notifications Management Jobs Instance Groups Instances Applications Execution Environments Topology View Settings

Templates > DG Preconfig Details

Back to Templates Details Access Notifications Schedules Jobs Survey

Name	DG Preconfig	Description	preconfig Primary and Standby DB for DG Setup	Job Type ⓘ	run
Organization	Default	Inventory ⓘ	DG Setup	Project ⓘ	DG Setup
Execution Environment ⓘ	PODBA Dataguard EE	Playbook ⓘ	dataguard_playbook.yml	Forks ⓘ	0
Verbosity ⓘ	0 (Normal)	Timeout ⓘ	0	Show Changes ⓘ	Off
Job Slicing ⓘ	1	Created	27/02/2025, 13:16:53 by admin	Last Modified	09/04/2025, 10:41:13 by admin
Credentials ⓘ	SSH: DG Setup				
Job Tags ⓘ	primary_config standby_config				
Variables ⓘ	YAML JSON	<pre>1- ##### 2 # This File contain all the Global Variables required to do the dataguard configuration for 19c database. 3 # Please update the variables by reading the comments provided at the 4 # beginning of each section.</pre>			
Edit Launch Delete					

Red Hat Ansible Automation Platform

Activity Stream Workflow Approvals Host Metrics

Resources ▼

- Templates
- Credentials
- Projects
- Inventories
- Hosts

Access ▼

- Organizations
- Users
- Teams

Administration ▼

- Credential Types
- Notifications
- Management Jobs
- Instance Groups
- Instances
- Applications
- Execution Environments
- Topology View

Settings

Templates > DG Restore Details

Back to Templates Details Access Notifications Schedules Jobs Survey

Name	DG Restore	Description	Standby Restore DB for DG Setup	Job Type	run
Organization	Default	Inventory	DG Setup	Project	DG Setup
Execution Environment	PODBA Dataguard EE	Playbook	dataguard_playbook.yml	Forks	0
Verbosity	0 (Normal)	Timeout	0	Show Changes	Off
Job Slicing	1	Created	27/02/2025, 13:17:30 by admin	Last Modified	09/04/2025, 10:45:52 by admin
Credentials	SSH: DG Setup				
Job Tags	create_standby				
Variables	YAML JSON	<pre>1- # Sample 4. ASM RMAN Duplicate 2 ##### 3 # This File contain all the Global Variables required to do the dataguard configuration for 19c database. 4 # Please update the variables by reading the comments provided at the</pre>			

Edit Launch Delete

Red Hat Ansible Automation Platform

Activity Stream Workflow Approvals Host Metrics

Resources ▼

- Templates
- Credentials
- Projects
- Inventories
- Hosts

Access ▼

- Organizations
- Users
- Teams

Administration ▼

- Credential Types
- Notifications
- Management Jobs
- Instance Groups
- Instances
- Applications
- Execution Environments
- Topology View

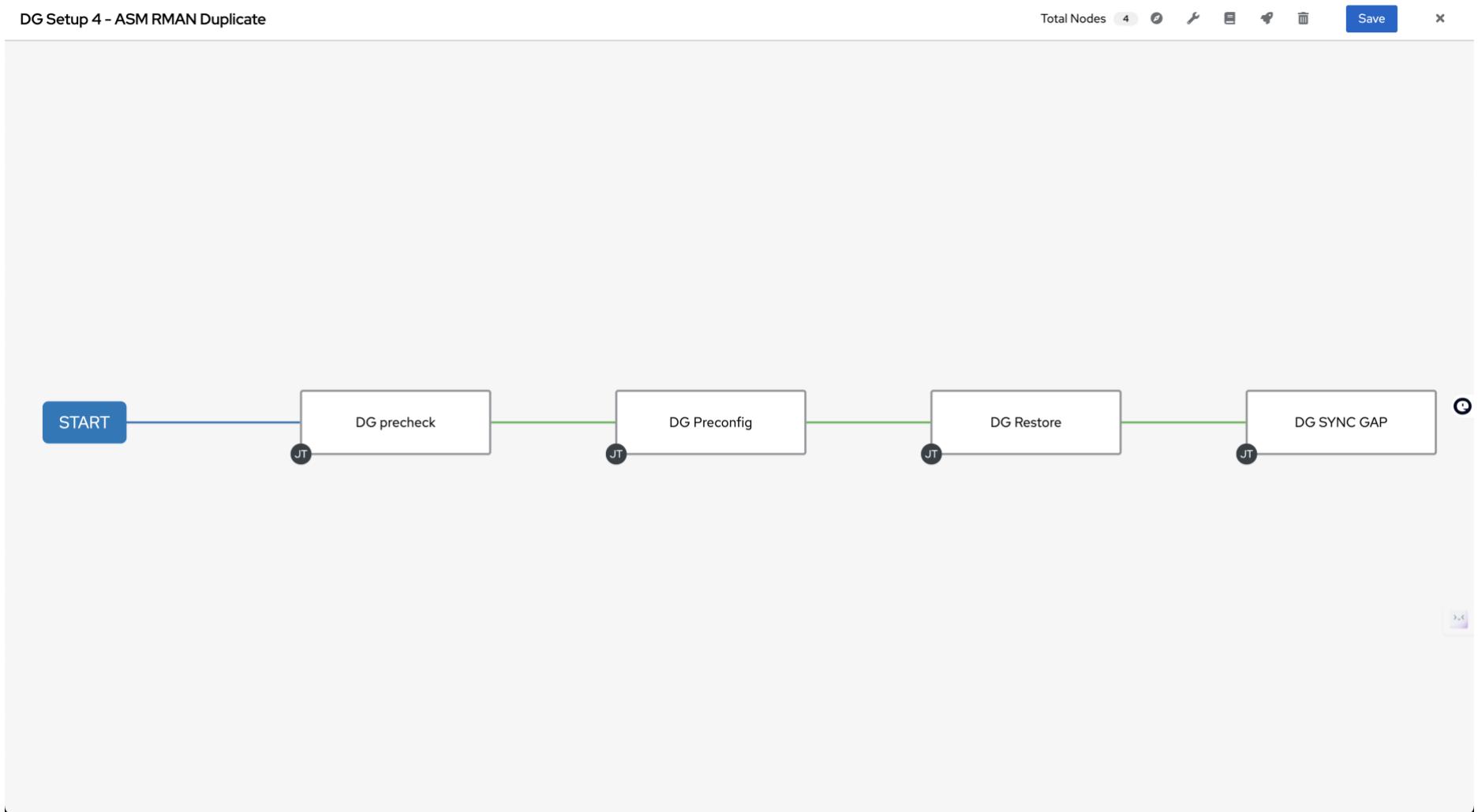
Settings

Templates > DG SYNC GAP Details

Back to Templates Details Access Notifications Schedules Jobs Survey

Name	DG SYNC GAP	Description	SYNC DB for DG Setup	Job Type	run
Organization	Default	Inventory	DG Setup	Project	DG Setup
Execution Environment	PODBA Dataguard EE	Playbook	dataguard_playbook.yml	Forks	0
Verbosity	0 (Normal)	Timeout	0	Show Changes	Off
Job Slicing	1	Created	27/02/2025, 13:21:46 by admin	Last Modified	09/04/2025, 10:47:19 by admin
Credentials	SSH: DG Setup				
Job Tags	post_sync_check				
Variables	YAML JSON	<pre>1- # Sample 4. ASM RMAN Duplicate 2 ##### 3 # This File contain all the Global Variables required to do the dataguard configuration for 19c database. 4 # Please update the variables by reading the comments provided at the</pre>			

Edit Launch Delete



Details

Name		Description	Activity
DG Setup 4 - ASM RMAN Duplicate		DG setup RMAN backup from external backups & Duplication from active database	✓ ! ! ! ! ✓ ! !
Job Type	Workflow Job Template	Inventory	DG Setup
Modified	06/04/2025, 15:49:36 by admin	Created	27/02/2025, 13:23:08 by admin
Variables			
<pre>1* # Sample 2. JFS2 RMAN Duplicate 2 ########## 3 # This File contain all the Global Variables required to do the dataguard configuration for 19c database. 4 # Please update the variables by reading the comments provided at the</pre>			
Edit Launch Delete			

9. Now launch the Job Template by clicking on “Launch”

Red Hat Ansible Automation Platform

Activity Stream

Workflow Approvals

Host Metrics

Resources ▼

- Templates DG Setup 4 - ASM RMAN Duplicate
- Credentials
- Projects
- Inventories
- Hosts

Access ▼

- Organizations
- Users
- Teams

Administration ▼

- Credential Types
- Notifications
- Management Jobs
- Instance Groups
- Instances
- Applications
- Execution Environments
- Topology View

Settings

Details

Back to Templates Details Access Notifications Schedules Visualizer Jobs Survey

Name: DG Setup 4 - ASM RMAN Duplicate Description: DG setup RMAN backup from external backups & Duplication from active database Activity: ✓ ! ! ! ! ! ✓ ! !

Job Type: Workflow Job Template Inventory: DG Setup Created: 27/02/2025, 13:23:08 by admin

Modified: 06/04/2025, 15:49:36 by admin

Variables: YAML JSON

```

1- # Sample 2. JFS2 RMAN Duplicate
2 #####
3 # This File contain all the Global Variables required to do the dataguard configuration for 19c database.
4 # Please update the variables by reading the comments provided at the
  
```

Edit Launch Delete

verify once it completes successfully.

Red Hat Ansible Automation Platform

Activity Stream

Workflow Approvals

Host Metrics

Resources ▼

- Templates Jobs > 2727 - DG Setup 4 - ASM RMAN Duplicate
- Credentials
- Projects
- Inventories
- Hosts

Access ▼

- Organizations
- Users
- Teams

Administration ▼

- Credential Types
- Notifications
- Management Jobs
- Instance Groups
- Instances
- Applications
- Execution Environments
- Topology View

Settings

Output

Back to Jobs Details Output

DG Setup 4 - ASM RMAN Duplicate ✓ Successful

Total Nodes: 4

```

graph LR
    START([START]) --> DGPrecheck[DG precheck  
00:00:46]
    DGPrecheck --> DGPreconfig[DG Preconfig  
00:00:40]
    DGPreconfig --> DGRestore[DG Restore  
00:01:32]
    DGRestore --> DGSyncGap[DG SYNC GAP  
00:03:36]
    
```

```

454 TASK [create_standby : Set fact for condition evaluation] *****
455 ok: [primary-db]
456
457 ok: [standby-db]
458
459 TASK [create_standby : Copy Protection Mode Post-Processing Script on primary] ***
460 skipping: [standby-db]
461 changed: [primary-db]
462
463 TASK [create_standby : Execute Protection Mode Post-Processing Script on primary] ***
464 skipping: [standby-db]
465 ok: [primary-db]
466
467 TASK [create_standby : Display Protection Mode Post-Processing Output] *****
468 ok: [primary-db] => {
469   "msg": [
470     "PROTECTION_MODE:MAXIMUM PERFORMANCE",
471     "Dataguard successfully configured with PROTECTION_MODE: MAXIMUM PERFORMANCE",
472     "Dataguard configured successfully"
473   ]
474 }
475 skipping: [standby-db]
476
477 PLAY RECAP *****
478 primary-db : ok=18 changed=2 unreachable=0 failed=0 skipped=14 rescued=0 ignored=0
479 standby-db : ok=21 changed=2 unreachable=0 failed=0 skipped=10 rescued=0 ignored=0

```

```

127
128 TASK [post_sync_check : Debug Raw SQL Output] *****
129 skipping: [primary-db]
130 ok: [standby-db] => {
131   "dg_sync_status.stdout_lines": [
132     "",
133     "      Thread Last Sequence Received Last Sequence Applied Difference",
134     "-----",
135     "\t 1\t\t\t 176\t\t\t 176 \t 0"
136   ]
137 }
138
139 TASK [post_sync_check : Extract Data Guard Sync Status] *****
140 skipping: [primary-db]
141 ok: [standby-db]
142
143 TASK [post_sync_check : Print Data Guard Sync Status] *****
144 skipping: [primary-db]
145 ok: [standby-db] => {
146   "msg": "#*Data Guard Sync Check**\n-----\n**Thread:** 1\n**Last Sequence Received:** 176\n**Last Sequence Applied:** 176\n**Difference:** 0\n-----\n**Status:** In Sync\n"
147 }
148
149 PLAY RECAP *****
150 primary-db : ok=19 changed=2 unreachable=0 failed=0 skipped=12 rescued=0 ignored=0
151 standby-db : ok=18 changed=2 unreachable=0 failed=0 skipped=10 rescued=0 ignored=0

```

10. If you want to execute only the precheck role, then along with the other details you add “job Tags”

Name: [Enter the name of your job]
 Description: [Enter a brief description of your job]
 Job Type: Run
 Inventory: [Select the inventory that was earlier created]
 Project: [Select the project that was earlier created]
 Execution Environment: [Select the execution environment that was earlier created]
 Playbook: [Enter the name of your playbook file]
 Variables: ansible_ssh_user: oracle
 Job Tags: dataguard_precheck
 After providing the required information, click on "Save".

Variables (YAML)

```

1- ##### This File contain all the Global Variables required to do the dataguard configuration for 19c database.
2 # Please update the variables by reading the comments provided at the
3 # beginning of each section.
4 # This variables file contain 4 sections:
5 #     A - Common Variables.

```

Forks (0) Job Slicing (1) Instance Groups (0)

Verosity (0 (Normal)) Show Changes (Off) Options

Privilege Escalation Provisioning Callbacks Enable Webhook Concurrent Jobs Enable Fact Storage Prevent Instance Group Fallback

Save **Cancel**

If you want to execute only SYNC between primary and standby, then along with the other details you add “job Tags”

Note: podba_rac_crs_upgrade role have a dependency on podba_rac_upgrade_precheck role

Name: [Enter the name of your job]

Description: [Enter a brief description of your job]

Job Type: Run

Inventory: [Select the inventory that was earlier created]

Project: [Select the project that was earlier created]

Execution Environment: [Select the execution environment that was earlier created]

Playbook: [Enter the name of your playbook file]

Variables: ansible_ssh_user: oracle

Job Tags: dataguard_post_sync_check

After providing the required information, click on "Save".

Variables (YAML)

```

1- # Sample 4. ASM RMAN Duplicate
2 #####
3 # This File contain all the Global Variables required to do the dataguard configuration for 19c database.
4 # Please update the variables by reading the comments provided at the
5 # beginning of each section.
6 # This variables file contain 4 sections:

```

Forks (0) Job Slicing (1) Instance Groups (0)

Verosity (0 (Normal)) Show Changes (Off) Options

Privilege Escalation Provisioning Callbacks Enable Webhook Concurrent Jobs Enable Fact Storage Prevent Instance Group Fallback

Save **Cancel**

12. 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-dba/issues>

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.