

PODBA using Ansible Automation Platform 2 (GUI)

Overview: Ansible Automation Platform Version 2 offers new features and components. This readme outlines the process of executing the PODBA playbooks using AAP2 GUI.

Reference to AAP2: <https://www.ansible.com/blog/introducing-ansible-automation-platform-2>

AAP2 installation instructions can be found here:

[Chapter 3. Installing Red Hat Ansible Automation Platform Red Hat Ansible Automation Platform 2.3 | Red Hat Customer Portal](#)

Prepare the Execution Environment: This is a one-time setup before executing playbooks from AAP2 GUI.

- Login to the Ansible controller with “awx” user.
- Install python version 3.8 or later [# dnf install python3]
- Install podman using dnf [# dnf install podman]
- Install ansible-builder [\$ pip install ansible-builder]
- Install ansible-navigator [\$ pip install ansible-navigator], this is not mandatory for GUI.
- In any present working directory, create a directory named "context".
- Download and extract Oracle Instant client software from Oracle site:
<https://www.oracle.com/database/technologies/instant-client/downloads.html>
- **Note:** For Linux on Power, click on “other platforms” in the above URL.
- Inside the "context" directory place the extracted oracle client software directory with the name "oracle_client".
- Create a file (example: create_podba.yml) with the following content.

```
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
    - RUN pip3 install wheel
    - RUN python3.9 -m pip install cx_Oracle --upgrade
    - RUN ln -s /usr/lib64/libnsl.so.2 /usr/lib64/libnsl.so.1
    - COPY oracle_client /oracle_client_sw
    - COPY ansible-automation-platform-managed-ca-cert.crt /etc/pki/ca-trust/source/anchors
dependencies:
  galaxy: requirements.yml
```

- Make the requirements.yml file

```
collections:
  - ibm.power_aix
```

- Run the following command to build the execution environment image.

```
$ ansible-builder build -t powerodba -f create_podba.yml Running
command:
podman build -f context/Containerfile -t powerodba context
Complete! The build context can be found at: /var/lib/awx/aap2/context
```

- List the images:

```
$ podman images
REPOSITORY                                TAG      IMAGE ID      CREATED      SIZE localhost/powerodba
latest      e04948d6013a About a minute ago 908 MB
registry.redhat.io/ansible-automation-platform-24/ee-supported-rhel8 latest    b2d26de2d8de 4 months ago
1.79 GB registry.redhat.io/ansible-automation-platform-24/ee-minimal-rhel8 latest    c239714e9480 4 months ago 380
MB
```

Execute playbooks from CLI using ansible-navigator:

1. Create a file called ansible-navigator.yml inside the {{ collection_name }}/playbooks directory with the following content.

```
$ cat ansible-navigator.yml
---
ansible-navigator:  execution-
environment:        enabled: True
image: powerodba:latest # Name of the REPOSITORY:TAG
```

2. Follow the readme files under "[docs](#)" to understand how to update the required variables for each task.
3. Run the following command to execute the playbooks. The following example shows execution of manage-db-directories.yml playbook.

```
ansible-navigator run <playbook name> --pp=missing --m stdout -i <name of inventory file>
Example:
$ ansible-navigator run manage-db-directories.yml --pp=missing --m stdout -i hosts.yml
```

To use the escalated privileges, please use "--playbook-artifact-enable false" at the end of the command.

Example:

```
ansible-navigator run db-opatch.yml --pp=missing --m stdout -i hosts.yml --ask-become-pass --
playbookartifact-enable false
```

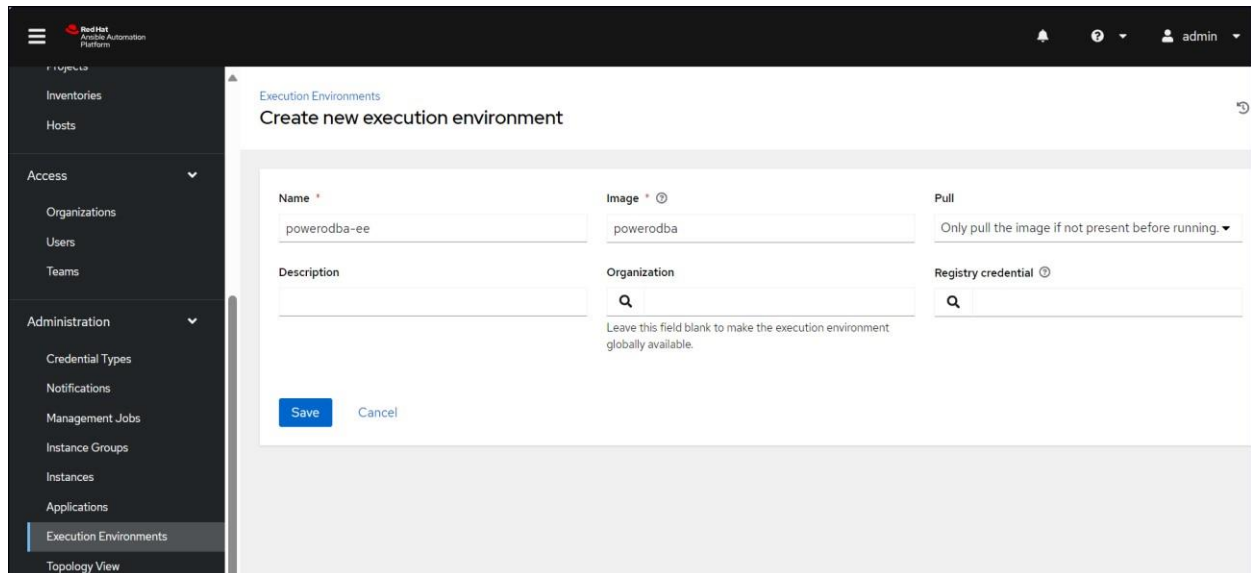
Execute the playbooks from GUI:

PODBA has two types of modules – few uses “cx_Oracle” module which is a python connector to access Oracle databases and the others won’t use cx_Oracle, they require “ssh” connectivity.

The playbook template setup will be a little different between the two types of modules. Hence, we’re going to setup a project in AAP2 and demonstrate one example playbook which uses cx_Oracle and another one which doesn’t require cx_Oracle.

Create Project:

Step 1: Login to AAP2 and “Create new execution environment”.



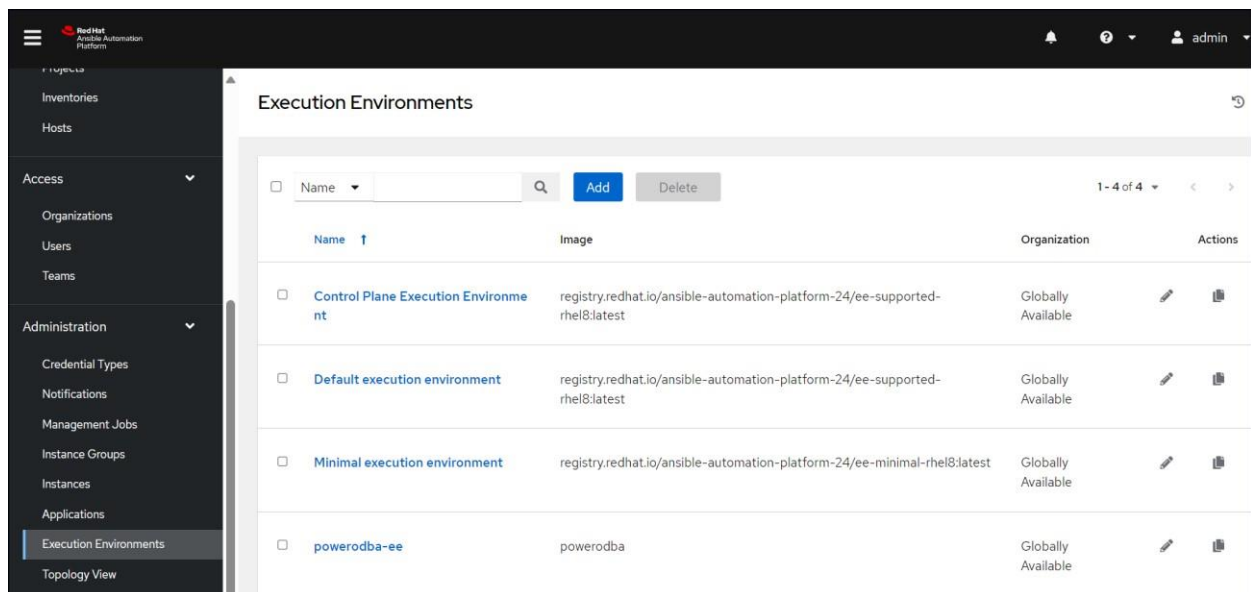
The screenshot shows the 'Create new execution environment' form in the AAP2 GUI. The form has the following fields:

- Name ***: powerodba-ee
- Image ***: powerodba
- Pull**: Only pull the image if not present before running. (dropdown)
- Description**: (empty text area)
- Organization**: (searchable dropdown)
- Registry credential**: (searchable dropdown)

Below the fields are 'Save' and 'Cancel' buttons. A note states: 'Leave this field blank to make the execution environment globally available.'

Name: Any desired name for identification.

Image: Provide the name of the image which was created earlier in this document using “ansible-builder”.



The screenshot shows the 'Execution Environments' list in the AAP2 GUI. The table has the following columns: Name, Image, Organization, and Actions. The table contains four entries:

Name	Image	Organization	Actions
Control Plane Execution Environment	registry.redhat.io/ansible-automation-platform-24/ee-supported-rhel8:latest	Globally Available	[Edit] [Delete]
Default execution environment	registry.redhat.io/ansible-automation-platform-24/ee-supported-rhel8:latest	Globally Available	[Edit] [Delete]
Minimal execution environment	registry.redhat.io/ansible-automation-platform-24/ee-minimal-rhel8:latest	Globally Available	[Edit] [Delete]
powerodba-ee	powerodba	Globally Available	[Edit] [Delete]

Step 2: Create a new project.

The screenshot shows the 'Create New Project' form in the Red Hat Ansible Automation Platform. The form is divided into several sections:

- Name:** A text input field containing 'powerodba'.
- Description:** An empty text input field.
- Organization:** A dropdown menu showing 'Default'.
- Execution Environment:** A dropdown menu showing 'powerodba-ee'.
- Source Control Type:** A dropdown menu showing 'Git'.
- Content Signature Validation Credential:** An empty text input field.
- Type Details:**
 - Source Control URL:** A text input field containing 'https://github.com/IBM/ansible-power-aix-oracle-'. There is a small icon to the right of the field.
 - Source Control Branch/Tag/Commit:** An empty text input field.
 - Source Control Refspec:** An empty text input field.
 - Source Control Credential:** An empty text input field.
- Options:** A section with several checkboxes:
 - ☐ Clean
 - ☐ Delete
 - ☐ Track submodules
 - ☐ Update Revision on Launch
 - ☐ Allow Branch Override

Name: Any desired name for identification.

Organization: Your existing Organization name or leave it “Default”

Execution Environment: Name of the Execution Environment created in Step 1 of **Create Project** section.

Source Control Type: Git

Source Control URL: <https://github.com/IBM/ansible-power-aix-oracle-dba>

Click “save”

The “Last Job Status” must show Successful as shown below.

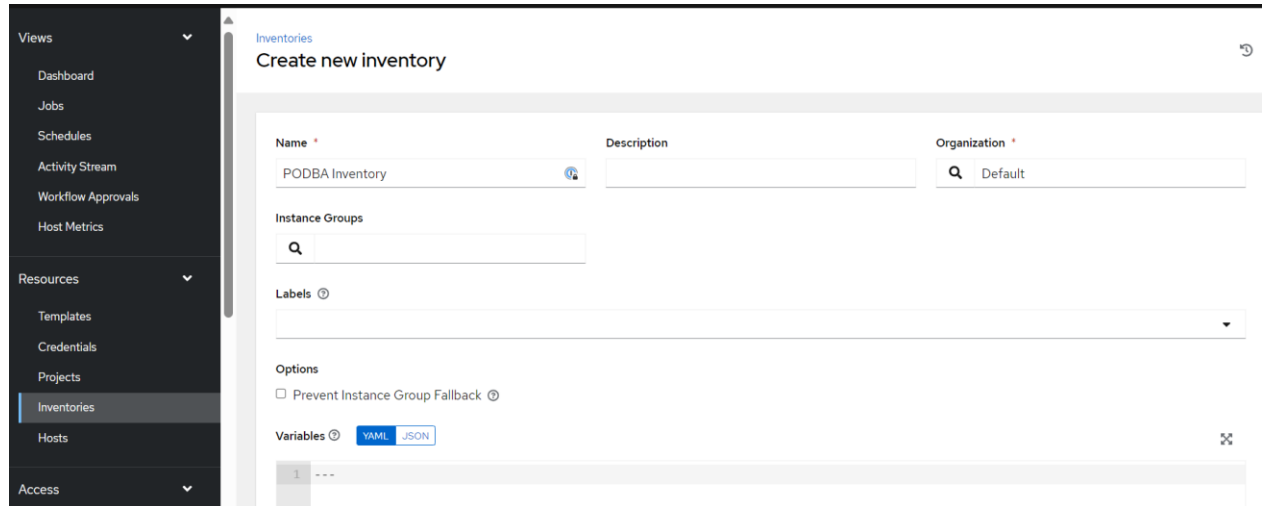
Projects > powerodba

Details

Back to Projects Details Access Job Templates Notifications Schedules					
Last Job Status	Successful	Name	powerodba	Organization	Default
Source Control Type	Git	Source Control Revision	a45c67a	Source Control URL	https://github.com/IBM/ansible-power-aix-oracle-dba
Cache Timeout	0 Seconds	Default Execution Environment	powerodba-ee	Project Base Path	/var/lib/awx/projects
Playbook Directory	_42__powerodba	Created	3/29/2024, 6:04:55 PM by admin	Last Modified	3/29/2024, 6:04:55 PM by admin
Edit Sync Delete					

Example 1 – Create Database : We are going to demonstrate how to create a Single Instance container database (CDB) named “testdb”, this database will reside on Automatic Storage Management (ASM).

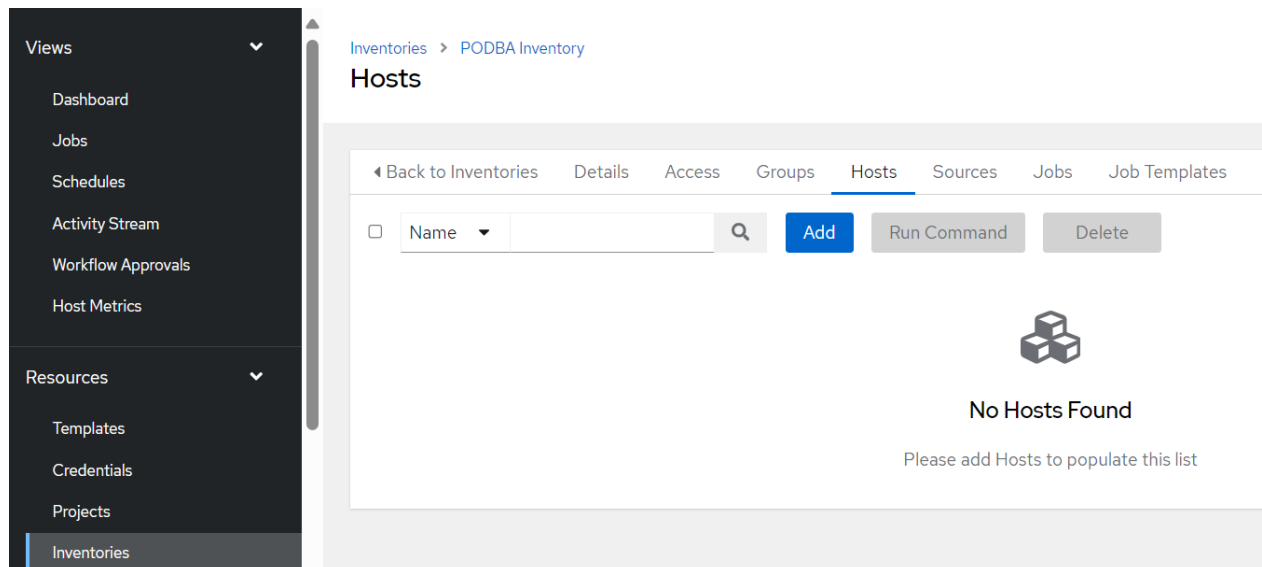
Step 1: Use an existing inventory or create a new one. In this case, I’m creating a new one.



The screenshot shows the 'Create new inventory' form in the Oracle Cloud console. The left sidebar contains a navigation menu with 'Views' (Dashboard, Jobs, Schedules, Activity Stream, Workflow Approvals, Host Metrics) and 'Resources' (Templates, Credentials, Projects, Inventories, Hosts, Access). The 'Inventories' section is selected. The main form has fields for 'Name' (PODBA Inventory), 'Description', and 'Organization' (Default). Below these are 'Instance Groups' and 'Labels' sections. The 'Options' section includes a checkbox for 'Prevent Instance Group Fallback'. The 'Variables' section has tabs for 'YAML' and 'JSON'. A table at the bottom shows a single row with the number '1'.

Name: PODBA Inventory (provide any name for identification) and Click on Save.

Step 2: Add the target lpar hostname where you need to create the database to the inventory.



The screenshot shows the 'Hosts' page for the 'PODBA Inventory'. The left sidebar is the same as in the previous screenshot. The main content area has a breadcrumb 'Inventories > PODBA Inventory' and a title 'Hosts'. Below the title is a navigation bar with tabs: 'Back to Inventories', 'Details', 'Access', 'Groups', 'Hosts' (selected), 'Sources', 'Jobs', and 'Job Templates'. Below the tabs is a search bar with a dropdown menu, a search icon, and buttons for 'Add', 'Run Command', and 'Delete'. The main area displays a message: 'No Hosts Found' with a cube icon and the text 'Please add Hosts to populate this list'.

Inventories > PODBA Inventory > Hosts

Create new host

Name * Description

Variables YAML JSON

1	---
2	ansible_host: 192.168.1.10

Save Cancel

Name: provide the hostname of the lpar along with the host IP address under the variables section and Click on Save.

Step 3: Setup the credentials of the lpar which we have added in the PODBA inventory in the previous step.

Credentials

Create New Credential

Name * Description Organization

Credential Type *

Type Details

Username Password ☐ Prompt on launch

Name: provide any name for identification and update the rest of the details as shown in the above fig. and Click on Save.

Step 4: Create a new job template for “Create DB” playbook.

The screenshot shows the 'Create New Job Template' interface. On the left is a sidebar with navigation links: Views (Dashboard, Jobs, Schedules, Activity Stream, Workflow Approvals, Host Metrics), Resources (Templates, Credentials, Projects, Inventories, Hosts), and Access. The main area is titled 'Create New Job Template' and contains the following fields:

- Name ***: Create SI DB
- Description**: (empty)
- Job Type ***: Run (with a 'Prompt on launch' checkbox)
- Inventory ***: PODBA Inventory (with a 'Prompt on launch' checkbox)
- Project ***: powerodba (with a 'Prompt on launch' checkbox)
- Execution Environment**: powerodba-ee (with a 'Prompt on launch' checkbox)
- Playbook ***: playbooks/create-dbyml
- Credentials**: SSH: ansible_db (with a 'Prompt on launch' checkbox)
- Labels**: (empty) (with a 'Prompt on launch' checkbox)
- Variables**: (empty) (with tabs for YAML and JSON, and a 'Prompt on launch' checkbox)

Name: Create SI DB (any name for identification). Update all the other details as shown in the above fig.

Variables: This section is an important one. Please refer the readme “Readme-Create-Database.txt” in the “docs” folder of this collection to understand more about each variable. I have used the following variables.

```
oracle_user: oracle
oracle_group: oinstall
oracle_stage: /tmp
oracle_base: /u01/app/oracle
oracle_dbf_dir_fs:
oracle_reco_dir_fs:
oracle_dbf_dir_asm: '+DATA1'
oracle_reco_dir_asm: '+DATA1'

oracle_databases:
- home: db1
  oracle_version_db: 19.3.0.0
  oracle_home: /u01/db19c_2
  oracle_edition: EE
  oracle_db_name: testdb
  oracle_db_type: SI
  is_container: True
  pdb_prefix: testpdb
  num_pdbs: 1
  storage_type: ASM
  service_name: testdb
  oracle_init_params: "sga_max_size=10g,sga_target=10g"
  oracle_db_mem_totalmb: 15000
  oracle_database_type: MULTIPURPOSE
  redolog_size_in_mb: 50
  state: present
```

Click on Save.

Step 5: After saving the template, create two surveys for prompting the “SYS password of ASM” and a new “SYS user password” during runtime.

The screenshot shows the 'Details' page for the 'Create SI DB' template. The left sidebar contains navigation links: Views, Resources (Templates, Credentials, Projects, Inventories, Hosts), and Access. The main content area has tabs for Back to Templates, Details (selected), Access, Notifications, Schedules, Jobs, and Survey. The Details tab displays the following information:

Field	Value
Name	Create SI DB
Job Type	run
Organization	Default
Inventory	PODBA Inventory
Project	powerodba
Execution Environment	powerodba-ee
Playbook	playbooks/create-db.yml
Forks	0
Verbosity	0 (Normal)
Timeout	0
Show Changes	Off
Job Slicing	1
Created	3/29/2024, 6:30:37 PM by admin
Last Modified	3/29/2024, 6:30:37 PM by admin
Credentials	SSH: ansible_db
Variables	YAML, JSON

Below the variables section, there is a list of variables:

```
1 ---
2 oracle_user: oracle
3 oracle_group: oinstall
4 oracle_stage: /tmp
```

The screenshot shows the 'Add Question' page for the 'Create SI DB' template. The left sidebar is the same as the previous screenshot. The main content area has tabs for Back to Templates, Details, Access, Notifications, Schedules, Jobs, and Survey. The Survey tab is selected, and the 'Add Question' page is displayed. The form contains the following fields:

Field	Value
Question *	Enter SYS ASM password
Description	
Answer variable name *	default_gipass
Answer type *	Password
Required	<input checked="" type="checkbox"/>
Minimum length	6
Maximum length	1024
Default answer	<input type="text"/>

At the bottom of the form, there are 'Save' and 'Cancel' buttons.

Question: Enter something to type the password, this will be prompted during runtime.

Answer variable name: Update “default_gipass”, this is explicitly set inside the roles.

Answer type: Password.

Click on Save.

Step 6: Create another survey for the new database's SYS user.

The screenshot shows the 'Add Question' form within the 'Create SI DB' survey template. The left sidebar contains 'Views' (Dashboard, Jobs, Schedules, Activity Stream, Workflow Approvals, Host Metrics) and 'Resources' (Templates, Credentials, Projects, Inventories). The form fields are as follows:

Question *	Description	Answer variable name * ⓘ
<input type="text" value="Enter a new SYS password for the database"/>	<input type="text"/>	<input type="text" value="default_dbpass"/>

Below the form, there are additional settings:

- Answer type *** ⓘ: Password (dropdown)
- ☒ Required
- Minimum length**:
- Maximum length**:
- Default answer**:

At the bottom, there are 'Save' and 'Cancel' buttons.

Click on Save.

Step 6: Make sure to toggle “Survey Enabled”.

The screenshot shows the 'Survey' tab in the 'Create SI DB' survey template. The left sidebar is the same as in the previous image. The top navigation bar includes 'Back to Templates', 'Details', 'Access', 'Notifications', 'Schedules', 'Jobs', and 'Survey' (which is selected). Below the navigation bar, there is a row of controls: a checkbox, an 'Add' button, an 'Edit Order' button, a 'Delete' button, and a 'Survey Enabled' toggle switch (which is turned on and highlighted with a yellow box). Below this is a table with the following columns: Name, Type, Default, and Action.

	Name	Type	Default	Action
<input type="checkbox"/>	Enter SYS ASM password *	password	ENCRYPTED	
<input type="checkbox"/>	Enter a new SYS password for the database *	password	ENCRYPTED	

Step 7: Launch the “Create SI DB” template

The screenshot shows the 'Details' page for the 'Create SI DB' template. The left sidebar contains navigation links: Views (Dashboard, Jobs, Schedules, Activity Stream, Workflow Approvals, Host Metrics), Resources (Templates, Credentials, Projects, Inventories, Hosts), and Access (Organizations). The main content area has tabs for Back to Templates, Details (selected), Access, Notifications, Schedules, Jobs, and Survey. The details section displays the following information:

Name	Job Type	Organization
Create SI DB	run	Default

Inventory	Project	Execution Environment
PODBA Inventory	powerodba	powerodba-ee

Playbook	Forks	Verbosity
playbooks/create-db.yml	0	0 (Normal)

Timeout	Show Changes	Job Slicing
0	Off	1

Created: 3/29/2024, 6:30:37 PM by admin
Last Modified: 3/29/2024, 6:41:46 PM by admin

Credentials: SSH: ansible_db

Variables: [YAML](#) [JSON](#)

```
1 ---
2 oracle_user: oracle
3 oracle_group: oinstall
4 oracle_stage: /tmp
```

Buttons: Edit, Launch, Delete

The dialog box is titled 'Launch | Create SI DB' and has a close button (X) in the top right corner. It contains a progress indicator on the left with two steps: 1 Survey (selected) and 2 Preview. The main area has two password input fields:

Enter SYS ASM password *

Enter a new SYS password for the database *

At the bottom, there are three buttons: Next (highlighted in blue), Back (disabled), and Cancel (disabled).

Input the passwords and click Next.

Launch | Create SI DB

1 Survey

2 Preview

Name	Create SI DB	Type	Job Template	Timeout	0 min 0 sec
Job Type	Run	Organization	Default	Inventory	PODBA Inventory
Project	powerodba	Execution Environment	powerodba-ee	Playbook	playbooks/create-db.yml
Forks	0	Verbosity	0 (Normal)	Show Changes	Off
Job Slicing	1				
Credentials	SSH: ansible_db				
Created	3/29/2024, 6:30:37 PM by admin	Last Modified	3/29/2024, 6:41:46 PM by admin		

Launch

Back

Cancel

Click on Launch.

The status of the job shows running along with the output.

Jobs > 349 - Create SI DB

Output

Back to Jobs

Details

Output

Create SI DB

Running

Elapsed 00:00:35

Stdout

Unfollow



'oracle_init_params': 'sga_max_size=10g,sga_target=10g', 'oracle_version_db': '19.3.0.0', 'pdb_prefix': 'testpdb', 'redo_log_size_in_mb': 50, 'service_name': 'testdb', 'state': 'present', 'storage_type': 'ASM'}, {'changed': True, 'end': '2024-03-29 06:16:20.090003', 'stdout': '0', 'cmd': 'grep testdb:/u01/db19c_2 /etc/oratab |wc -l', 'rc': 0, 'start': '2024-03-29 06:16:20.019296', 'stderr': '', 'delta': '0:00:00.070707', 'invocation': {'module_args': {'executable': None, 'uses_shell': True, 'strip_empty_ends': True, '_raw_params': 'grep testdb:/u01/db19c_2 /etc/oratab |wc -l', 'removes': None, 'argv': None, 'creates': None, 'chdir': None, 'stdin_add_newline': True, 'stdin': None}}, 'msg': '', 'stdout_lines': ['0'], 'stderr_lines': [], 'failed': False, 'item': {'home': 'db1', 'is_container': True, 'num_pdb': 1, 'oracle_database_type': 'MULTIPURPOSE', 'oracle_db_mem_totalmb': 15000, 'oracle_db_name': 'testdb', 'oracle_db_type': 'SI', 'oracle_edition': 'EE', 'oracle_home': '/u01/db19c_2', 'oracle_init_params': 'sga_max_size=10g,sga_target=10g', 'oracle_version_db': '19.3.0.0', 'pdb_prefix': 'testpdb', 'redo_log_size_in_mb': 50, 'service_name': 'testdb', 'state': 'present', 'storage_type': 'ASM'}, 'ansible_loop_var': 'item'}}]


40

41 TASK [oradb_create : Create database(s)] ***** 18:46:21

Output

[Back to Jobs](#) [Details](#) [Output](#)

Create SI DB ✓ Successful Plays 1 Tasks 14 Hosts 1 Elapsed 00:19:27  

Stdout 

86

87 TASK [oradb_create : debug] ***** 19:05:26

88 ok: [ansible_db] => {

89 "psout.stdout_lines": [

90 " oracle 16384508 1 0 Mar 18 - 0:06 asm_pmon_+ASM",

91 " oracle 18940250 1 0 Mar 18 - 0:12 ora_pmon_atstdb",

92 " oracle 53084600 1 0 06:35:02 - 0:00 ora_pmon_testdb"

93]

94 }

95

96 PLAY RECAP ***** 19:05:27

97 ansible_db : ok=11 changed=3 unreachable=0 failed=0 skipped=3 rescued=0 ignored=0

DB creation successful.

Example 2 – Create DB users: We are going to demonstrate how to create two DB users namely testuser1 and testuser2 in a PDB called testpdb.

Step 1: Create a new job template for “Create User” playbook.

The screenshot shows the 'Create New Job Template' interface. On the left is a sidebar with 'Views' and 'Resources' sections. The 'Resources' section has 'Templates' selected. The main form has the following fields:

- Name:** Create DB User
- Description:** (empty)
- Job Type:** Run
- Inventory:** PODBA Inventory
- Project:** powerodba
- Execution Environment:** powerodba-ee
- Playbook:** playbooks/manage-users.yml
- Credentials:** (empty)
- Labels:** (empty)

Each field has a search icon and a 'Prompt on launch' checkbox.

Step 2: Please refer the readme “Readme-Manage-Users.txt” in the “docs” folder of this collection to understand more about each variable. I have used the following variables.

```
hostname: ansible_db
listener_port: 1521
oracle_db_home: /oracle_client_sw
```

```
oracle_databases:
  - users:
    - schema: testuser1
      default_tablespace: users
      service_name: testpdb
      schema_password: oracle4u
      grants_mode: enforce
      grants:
        - connect
        - resource
      state: present
  - users:
    - schema: testuser2
      default_tablespace: users
      service_name: testpdb
      grants_mode: enforce
      grants:
        - connect
      schema_password: oracle4u
      state: present
```

Click on Save.

Step 3: After saving the template, create a survey for prompting the “SYS user password” during runtime.

Views

Dashboard
Jobs
Schedules
Activity Stream
Workflow Approvals
Host Metrics

Resources

Templates
Credentials
Projects
Inventories
Hosts

Access

Organizations

Templates > Create DB User

Details

Back to Templates
Details
Access
Notifications
Schedules
Jobs
Survey

Name	Create DB User	Job Type	run	Organization	Default
Inventory	PODBA Inventory	Project	powerodba	Execution Environment	powerodba-ee
Playbook	playbooks/manage-users.yml	Forks	0	Verbosity	0 (Normal)
Timeout	0	Show Changes	Off	Job Slicing	1
Created	3/29/2024, 7:03:28 PM by admin	Last Modified	3/29/2024, 7:03:28 PM by admin		

Variables
YAML
JSON

```

1 ---
2 hostname: ansible_db
3 listener_port: 1521
4 oracle_db_home: /oracle_client_sw

```

Edit
Launch
Delete

Views

Dashboard
Jobs
Schedules
Activity Stream
Workflow Approvals
Host Metrics

Resources

Templates
Credentials
Projects
Inventories

Templates > Create DB User > Survey

Add Question

Question *

Enter the SYS user password

Description

Answer variable name *

default_dbpass

Answer type *

Password

Required

☒

Minimum length

6

Maximum length

1024

Default answer

Save

Cancel

Question: Enter something to type the password, this will be prompted during runtime.

Answer variable name: Update “default_dbpass”, this is explicitly set inside the roles.

Answer type: Password.

Click on Save.

Make sure to toggle “Survey Enabled”.

Views

DashboardJobsSchedulesActivity StreamWorkflow ApprovalsHost MetricsResources

Templates > Create DB User

Survey

Back to TemplatesDetailsAccessNotificationsSchedulesJobsSurvey

☐

AddDelete

☒ Survey Enabled

Name	Type	Default	Actions
<input type="checkbox"/> Enter the SYS user password *	password	ENCRYPTED	

Launch the “Create Users” template.

Templates > Create DB User

Details

Back to TemplatesDetailsAccessNotificationsSchedulesJobsSurvey

Name	Create DB User	Job Type	run	Organization	Default
Inventory	PODBA Inventory	Project	powerodba	Execution Environment	powerodba-ee
Playbook	playbooks/manage-users.yml	Forks	0	Verbosity	0 (Normal)
Timeout	0	Show Changes	Off	Job Slicing	1
Created	3/29/2024, 7:03:28 PM by admin	Last Modified	3/29/2024, 7:09:52 PM by admin		

Variables

YAMLJSON

1

2

hostname: ansible_db

3

listener_port: 1521

4

oracle_db_home: /oracle_client_sw

Edit

Launch

Delete

Launch | Create DB User



1 Survey

2 Preview

Enter the SYS user password *

.....

Next

Back

Cancel

Input the SYS user password and click next.

Launch | Create DB User

1 Survey

2 Preview

Name	Create DB User	Type	Job Template	Timeout	0 min 0 sec
Job Type	Run	Organization	Default	Inventory	PODBA Inventory
Project	powerodba	Execution Environment	powerodba-ee	Playbook	playbooks/manage-users.yml
Forks	0	Verbosity	0 (Normal)	Show Changes	Off
Job Slicing	1	Created	3/29/2024, 7:03:28 PM by admin	Last Modified	3/29/2024, 7:09:52 PM by admin

Prompted Values

Variables

YAMLJSON

Launch

Back

Cancel

Click on Launch

Jobs > 350 - Create DB User

Output

Back to Jobs

Details

Output

Create DB User

Successful

Plays 1Tasks 3Hosts 1Elapsed 00:00:13

Stdout

3TASK [oradb_manage_users : Manage users (cdb/pdb)] ***** 19:13:54

4changed: [localhost] => (item=port: 1521 service: testpdb schema: testuser1 state:present)

5changed: [localhost] => (item=port: 1521 service: testpdb schema: testuser2 state:present)

6[WARNING]: Module did not set no_log for update_password

7

8TASK [oradb_manage_grants : Manage role grants] ***** 19:13:57

9skipping: [localhost]

10

11TASK [oradb_manage_grants : Manage schema grants] ***** 19:13:57

12changed: [localhost] => (item=port: 1521, service: testpdb, schema: testuser1, grants: ['connect', 'resource'], state: present)

13changed: [localhost] => (item=port: 1521, service: testpdb, schema: testuser2, grants: ['connect'], state: present)

14

15PLAY RECAP ***** 19:14:03

16localhost : ok=2 changed=2 unreachable=0 failed=0 skipped=1 rescued=0 ignored=0

User creation and grants successful.