

## Execute PODBA using Ansible Automation Platform 2 (AAP2)

**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 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
```

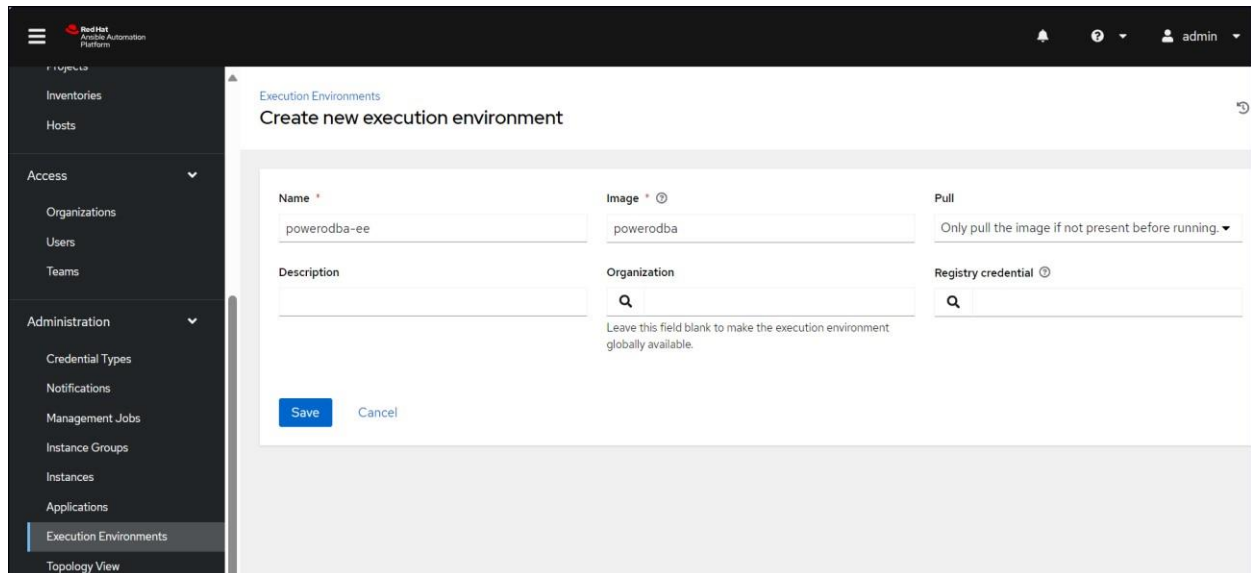
## Execute playbooks from GUI:

PODBA has two types of modules – few of them use “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”.



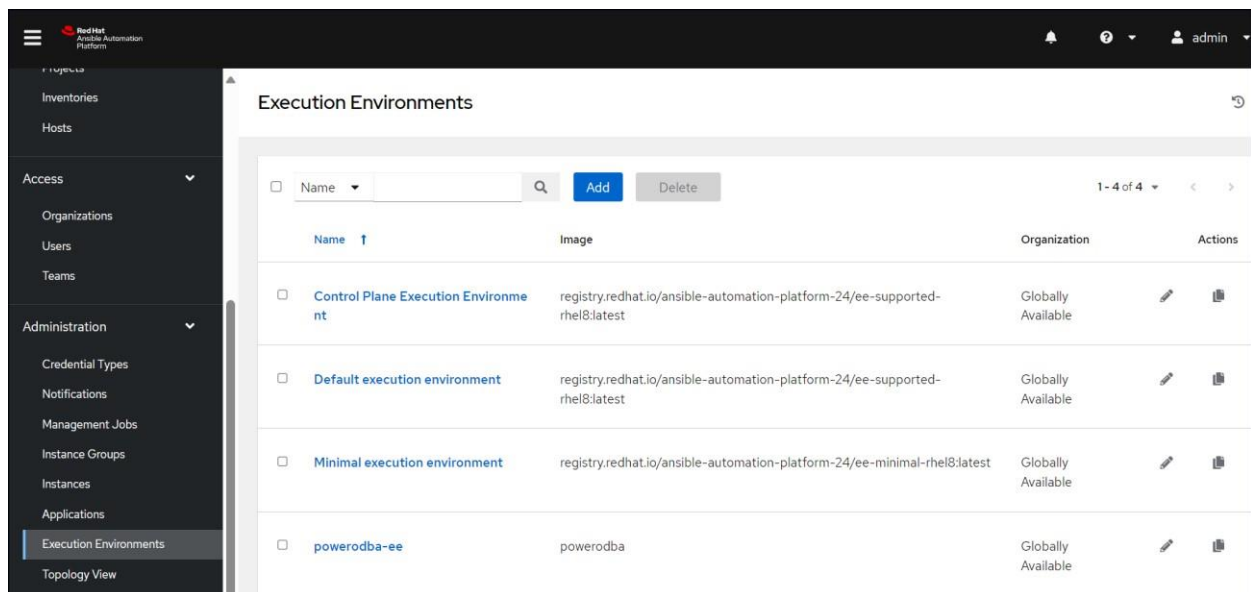
The screenshot shows the 'Create new execution environment' form in the AAP2 GUI. The form has the following fields:

- 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 left sidebar contains navigation links: Schedules, Activity Stream, Workflow Approvals, Host Metrics, Resources (Templates, Credentials, Projects, Inventories, Hosts), Access (Organizations, Users, Teams), and Administration. The main form area is titled 'Create New Project' and contains the following fields:

- Name \***: powerodba-project
- Description**: (empty)
- Organization \***: Default
- Execution Environment ⓘ**: powerodba-ee
- Source Control Type \***: Git
- Content Signature Validation Credential ⓘ**: (empty)
- Type Details**
  - Source Control URL \*** ⓘ: https://github.com/IBM/ansible-power-aix-oracle-i
  - Source Control Branch/Tag/Commit ⓘ**: (empty)
  - Source Control Refspec ⓘ**: (empty)
  - Source Control Credential**: (empty)
- Options**
  - ☐ 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.

**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.

The screenshot shows the 'Details' page for the 'powerodba-project' in the Red Hat Ansible Automation Platform. The left sidebar is the same as in the previous screenshot. The main area shows the project details with tabs for Back to Projects, Details, Access, Job Templates, Notifications, and Schedules. The 'Details' tab is active, showing the following information:

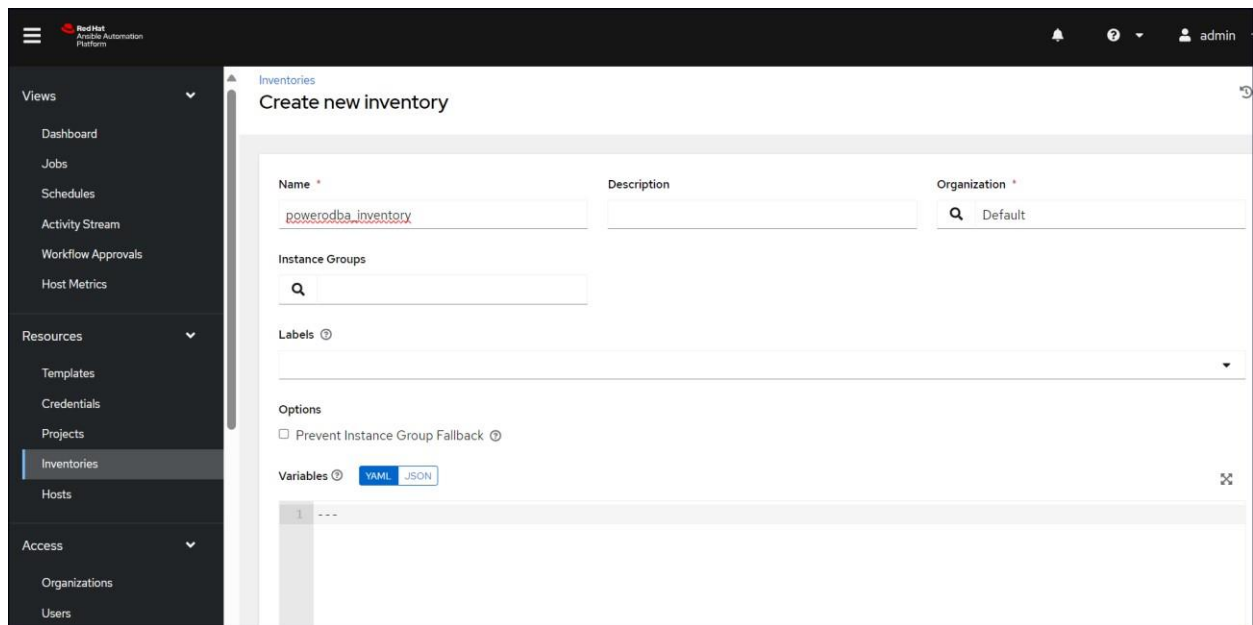
<b>Last Job Status</b>	Successful	<b>Name</b>	powerodba-project	<b>Organization</b>	Default
<b>Source Control Type</b>	Git	<b>Source Control Revision</b>	96a6db1	<b>Source Control URL ⓘ</b>	https://github.com/IBM/ansible-power-aix-oracle-dba
<b>Cache Timeout</b>	0 Seconds	<b>Default Execution Environment ⓘ</b>	powerodba-ee	<b>Project Base Path ⓘ</b>	/var/lib/awx/projects
<b>Playbook Directory ⓘ</b>	_i8__powerodba_project	<b>Created</b>	10/25/2023, 11:05:35 PM by admin	<b>Last Modified</b>	10/25/2023, 11:05:35 PM by admin

At the bottom of the details section are three buttons: Edit, Sync, and Delete.

From the above fig, the collection is installed in “/var/lib/awx/projects”

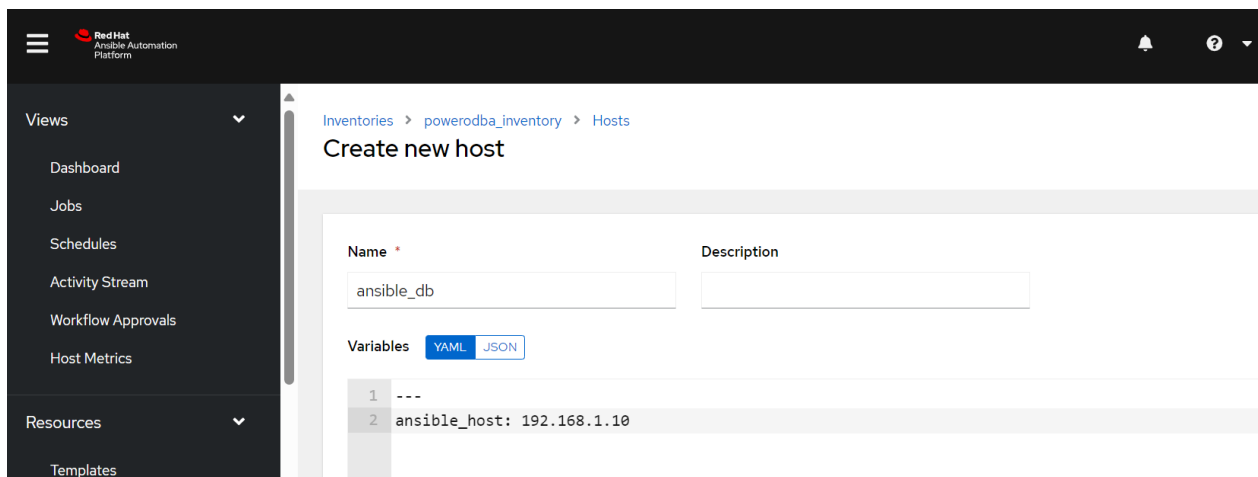
**Example 1:** The following setup is for the playbooks which use cx\_Oracle. The “manage-users.yml” playbook will create two users testuser1 and testuser2 in a Non-CDB environment “atsdb”.

**Step 1:** Create an inventory with a desired name and save it.



The screenshot shows the 'Create new inventory' form in the Red Hat Ansible Automation Platform. 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, Users). The main form has the following fields: Name (powerodba\_inventory), Description (empty), Organization (Default), Instance Groups (empty), Labels (empty), Options (Prevent Instance Group Fallback checkbox), and Variables (YAML/JSON tabs). The Variables section shows a single line with a comment: 1 ---

**Step 2:** Add the hostname where the database is running in the inventory and save it.



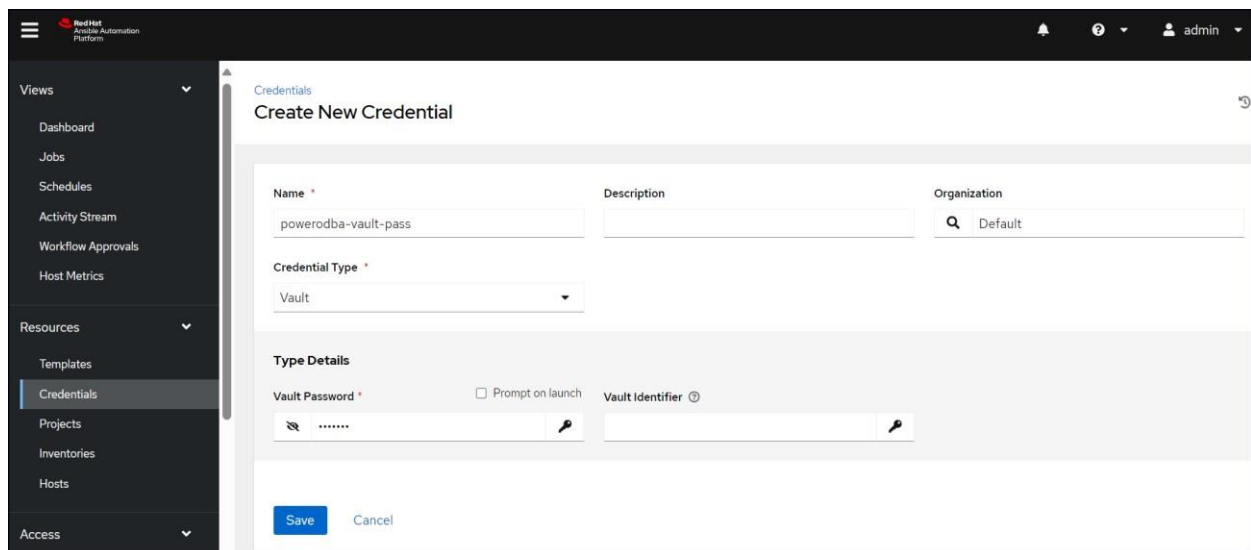
The screenshot shows the 'Create new host' form in the Red Hat Ansible Automation Platform. The left sidebar is the same as in the previous screenshot. The main form has the following fields: Name (ansible\_db), Description (empty), and Variables (YAML/JSON tabs). The Variables section shows two lines: 1 --- and 2 ansible\_host: 192.168.1.10. The breadcrumb navigation at the top of the form indicates the path: Inventories > powerodba\_inventory > Hosts.

**Step 3:** Set SYS user password in the vault.yml file for “default\_dbpass” variable and encrypt it.

```
[awx@localhost vars]$ pwd
/var/lib/awx/projects/_18__powerodba_project/playbooks/vars
[awx@localhost vars]$ cat vault.yml default_gipass:
Oracle4u      # ASM sys user password
default_dbpass: Oracle4u      # Sys user password

[awx@localhost vars]$ ansible-vault encrypt vault.yml
New Vault password:
Confirm New Vault password:
Encryption successful
```

**Step 4:** Set the vault password in “Create New Credentials” tab.



**Name:** Any desired name for identification.

**Credential Type:** vault

**Vault Password:** Update the password used in Step 5.

**Step 5:** Create a template for manage-users.yml playbook, this will create database users and grants privileges to it.

The screenshot shows the Red Hat Ansible Automation Platform interface. On the left is a sidebar with navigation options: Views (Dashboard, Jobs, Schedules, Activity Stream, Workflow Approvals, Host Metrics), Resources (Templates, Credentials, Projects, Inventories, Hosts), and Access (Organizations, Users). The main panel is titled 'Manage DB Users' and contains the following fields:

- Name:** Manage DB Users
- Description:** (empty)
- Job Type:** Run
- Inventory:** powerodba\_inventory
- Project:** powerodba-project
- Execution Environment:** powerodba-ee
- Playbook:** playbooks/manage-users.yml
- Credentials:** Vault: powerodba-va...
- Labels:** (empty)
- Variables:** (YAML/JSON tabs, showing a YAML snippet)
 

```

1 ---
2 hostname: ansible_db           # AIX lpar hostname where the database is running.
3 listener_port: 1522           # Database port number.
4 oracle_db_home: /oracle_client_sw # Oracle Instant Client path in the execution environment, refer Step 7 of "Prepare the
5
6 oracle_databases:              # Database users List to be created
7
      
```

**Name:** Manage DB Users

**Inventory:** Select the inventory created in Step 3 of “Create Project” section.

**Project:** Select the project created in Step 2 for “Create Project” section.

**Execution Environment:** Select the Execution environment used in Step 1

**Playbook:** From the drop down, select manage-users.yml

**Credentials:** Select the vault credentials created earlier.

The 'Select Credentials' dialog box shows a search for vault credentials. It includes a message: "You cannot select multiple vault credentials with the same vault ID. Doing so will automatically deselect the other with the same vault ID." The 'Selected Category' is set to 'Vault'. The 'Selected' list shows 'Vault: powerodba-va...'. Below is a search bar with the text 'Name' and a search icon. A table lists the available credentials:

Name	Selected
powerodba-vault-pass	<input checked="" type="checkbox"/>

At the bottom, there are 'Select' and 'Cancel' buttons. The page number '1 of 1 page' is displayed.

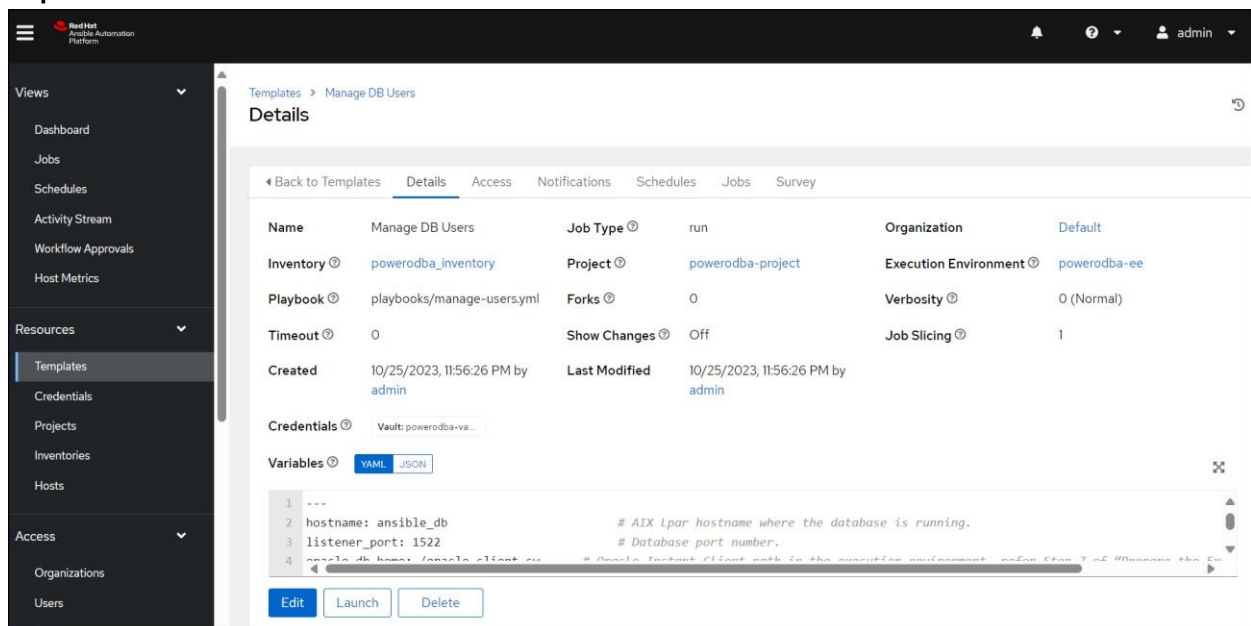
## Variables:

```
hostname: ansible_db                                # AIX lpar hostname where the database is running.
listener_port: 1521                                # Database port number.
oracle_db_home: /oracle_client_sw                  # Oracle Instant Client path on the ansible controller / execution environment.

oracle_databases:                                  # Database users list to be created
- users:
  - schema: testuser1                              # Username to be created.
    default_tablespace: users                       # Default tablespace to be assigned to the user.
    service_name: devpdb                            # Database service name.
    schema_password: oracle3                       # Password for the user.
    grants_mode: enforce                           # enforce|append.
    grants:
      - connect                                     # Provide name of the privilege as a list to grant to the user.
      - resource
    state: present                                # present|absent|locked|unlocked [present: Creates user, absent: Drops user]
# Multiple users can be created with different attributes as shown below.
- users:
  - schema: testuser2
    default_tablespace: users
    service_name: devpdb
    grants_mode: enforce
    grants:
      - connect
    schema_password: oracle4
    state: present
```

Save the template.

## Step 6: Click Launch

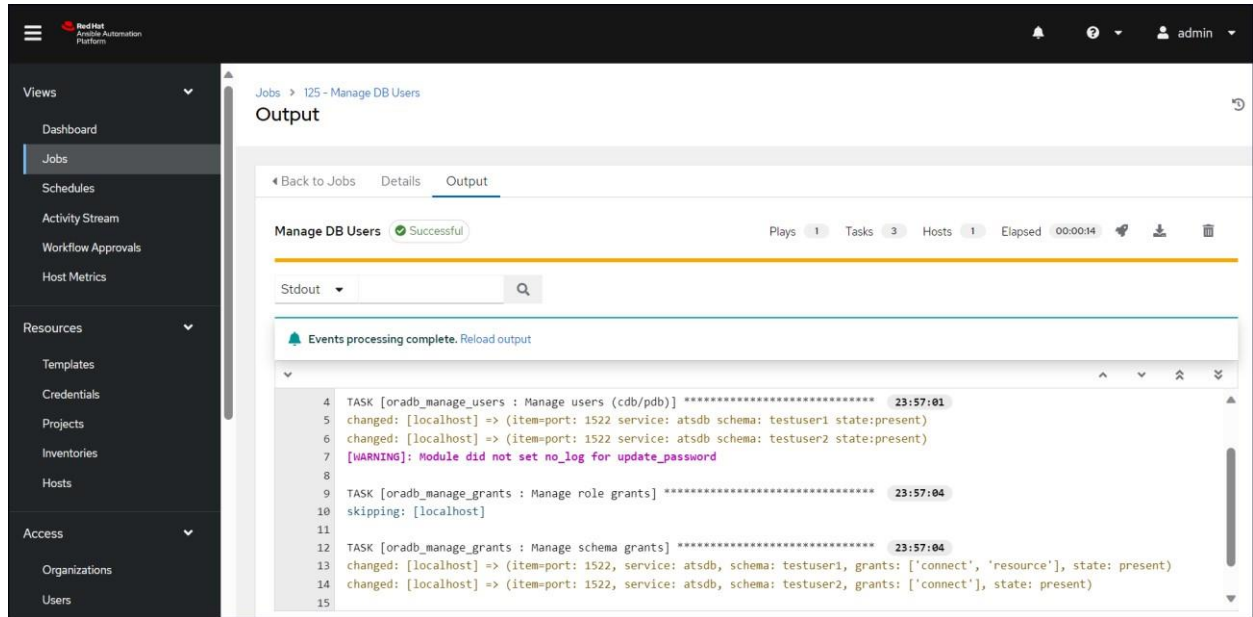


The screenshot shows the Red Hat Ansible Automation Platform interface. The left sidebar contains navigation options: Views (Dashboard, Jobs, Schedules, Activity Stream, Workflow Approvals, Host Metrics), Resources (Templates, Credentials, Projects, Inventories, Hosts), and Access (Organizations, Users). The main panel displays the 'Manage DB Users' template details. The template is named 'Manage DB Users', has a job type of 'run', and is associated with the 'powerodba-project' organization. The inventory is 'powerodba\_inventory', the playbook is 'playbooks/manage-users.yml', and the timeout is 0. The template was created on 10/25/2023 at 11:56:26 PM by 'admin'. The variables section shows the following YAML code:

```
---
hostname: ansible_db                                # AIX lpar hostname where the database is running.
listener_port: 1522                                # Database port number.
oracle_db_home: /oracle_client_sw                  # Oracle Instant Client path on the ansible controller / execution environment.
```

At the bottom of the variables section, there are buttons for 'Edit', 'Launch', and 'Delete'.





We can see two users testuser1 and testuser2 have been created and granted privileges.