

Contents

Versions	2
Description	2
Steps	4
Troubleshooting	11
Playbook Roles	11
Executing Oracle collection using Ansible Automation Platform 2 (AAP2).....	13
PowerVC – Infrastructure Provisioning	26
Appendix.....	30

This Ansible playbook installs Oracle database 19c Single Instance Database on AIX operating system, supports both JFS and Oracle ASM. Tested playbook on AIX 73 and PowerVS AIX partition. Also tested playbook using Ansible Automation Platform 2, you can find the steps in below sections.

Versions

1.2.1

Added powervc role “powervc_create_aixvm” which automates the creation of infrastructure needed for oracle deployment

1.2.0

- Multiple software staging options (nfs,remote(ansible-controller),local(managed host))
- ignoreprechecks option for oracle installers (gridSetup.sh and runInstaller)
- NFS support for installing filesets (optional)
- Bug fix on disk_validate.sh, RPM install skip if exists
- JFS redo blk size 512, tnsnames.ora entry for newly created DB
- Added cluvfy check for grid install (default), disable it using --skip-tags=runcuvfy
- Role separation for Grid and DB homes
- RU apply option during the install, supports 19.4 or later

1.1.1

- Added support for AIX 73, PowerVS and AAP2 execution

1.1.0

- Supports 19c DB install on JFS2 and ASM

1.0.0

- Supports 19c DB install on JFS2 only

Description

This playbook assumes the following:

- That the user is familiar with Ansible and should have basic knowledge on YAML, for the purpose of running this playbook
- That the user is familiar with Oracle Database Configuration
- That the user is familiar with the AIX Operating system
- That the version of AIX is 7.2 TL4 SP1 or later. (It should work on other versions of AIX supported by the oracle database AIX OS requirements, but has not been tested).
- That the playbook assumes a ****New AIX LPAR**** for execution
- That the targeted AIX LPAR for installing the Oracle single instance database will be referred within the rest of the document as the 'host' or 'managed host'.
- That the version of Oracle Standalone Database tested is 19.3.0.0 + RU (Optional)
- Depends on ibm.power_aix collection.

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>

<https://www.ibm.com/support/pages/oracle-db-rac-19c-ibm-aix-tips-and-considerations>

To get started with AIX refer

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

System Configuration

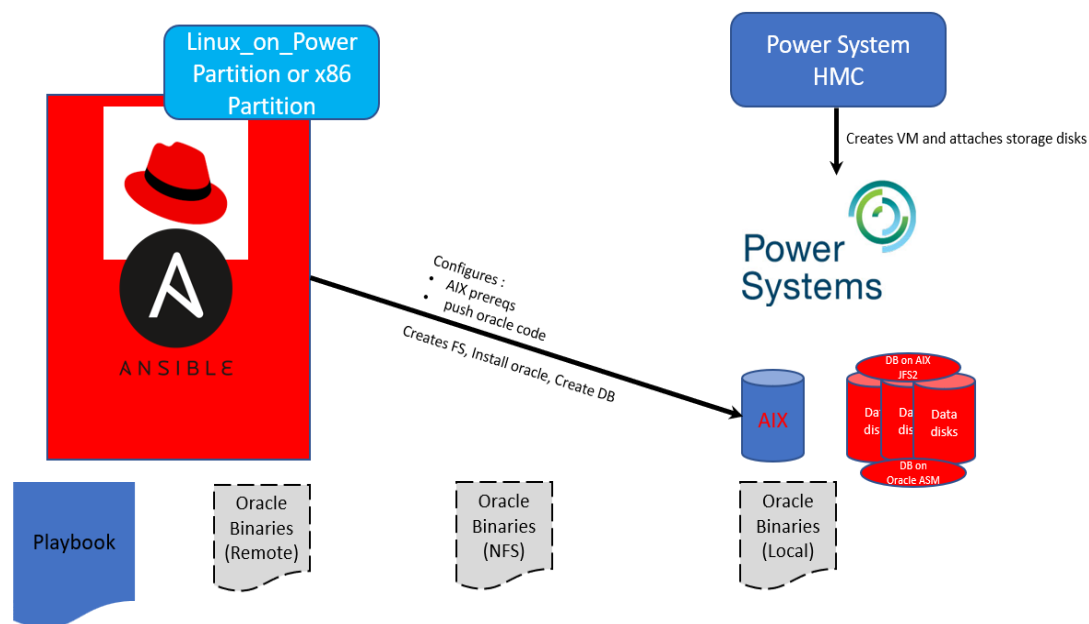


Figure. System Topology

The above snippet shows the high level system topology. The Ansible controller node can be on x86 or Linux on Power or any other system. The oracle binary zip files can be placed on NFS or Remote on Ansible Controller or Local on managed Host.

We have used two servers, one Linux_on_Power server used for running Ansible Controller and second one managed host AIX server used for installation and configuration on oracle 19c Database software.

a) Linux_on_Power server :

Operating System : RHEL 8.8

Ansible Engine Version : 2.15.0

For Ansible Controller prerequisites refer to below link

https://docs.ansible.com/ansible/latest/installation_guide/index.html

b) AIX server :

Operating System : AIX73TL1SP03

Oracle DB Version : 19.22.0.0

CPUs : 4

RAM : 64GB

Refer below link to get details of minimum software/hardware requirements that are need to run oracle 19c database on AIX operating system

<https://docs.oracle.com/en/database/oracle/oracle-database/19/axdbi/oracle-database-installation-checklist.html>

The collection contains below three roles

- **preconfig** : This role will perform AIX configuration tasks that are needed for oracle installation
- **oracle_install** : This role performs oracle binary installation
- **oracle_createdb** : This role creates test database “orcl” using dbca utility

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

Steps

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.

2. Setup ssh Equivalence with managed host(AIX) 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
```

3. Preparing to run the oracle playbook

3.1) The Oracle ansible collection depends on power_aix collection. The oracle collection uses ibm.power_aix collection modules like filesystem, devices, lvg and mount. Install latest version of ibm.power_aix collection from galaxy if it is not present

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

3.2) Download the Oracle single instance 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_ais_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

3.3) Update variables in variable file of power_ais_oracle collection, <collection dir>/vars/oracle_params.yml

3.3.1)

Download the Oracle 19c software from OTN or Oracle edelivery site

<https://edelivery.oracle.com/osdc/faces/SoftwareDelivery>

<https://www.oracle.com/database/technologies/oracle19c-ai-193000-downloads.html>

If you want to apply the Release Update patch during installation, Download the patch and latest Opatch from Oracle support site <https://support.oracle.com>

The oracle binary zip files can be placed on NFS or Remote on Ansible Controller or Local on managed Host. Update the variable ora_binary_location with nfs or remote or local and then update variables oracledbaix19c and oraclegridaix19c(optional, only needed when you are using Oracle ASM)

Below snippet shows nfs option for oracle binaries(both grid and DB) and applying the 19.22 RU version.

```
ora_binary_location: nfs

# Set the below if the variable "ora_binary_location=nfs"
ora_nfs_host: 192.168.127.10
ora_nfs_device:
  - /repos
ora_nfs_filesystem:
  - /binora

# Set the below locations as per the above "ora_binary_location" type
oracledbaix19c:
  - /binora/images/oracle/19c/V982583-01_193000_db.zip
oraclegridaix19c:
  - /binora/images/oracle/19c/V982588-01_193000_grid.zip
# Set below variables when using applyRU along with installation
ru_path: /binora/images/oracle/19c/RU19.22/p35940989_190000_AIX64-5L_RU19.22.zip
ru_version: 19.22
opatch_path: /binora/images/oracle/opatch/12.2.0.1.41/opatch-
12.2.0.1.41_p6880880_210000_AIX64-5L.zip
```

3.3.2) Next update the variables for RPMs and Filesets installation. NFS for filesets is optional and you keep these variables as empty if you are not using it

Below is the example snippet

```
preconfig:
  linux:
    tools: ['unzip']
#  tools: ['bash', 'unzip', 'sudo']
    optional_tools: []

# Provide optional nfs details for filesets installation
nfsfilesetMountPoint: /filesetaix
nfsfilesetExportDir: /nim/AIX73TL1/lppsource/installp/ppc
nfsfilesetServerIP: 192.168.76.16

# If nim filesystem mounted on target host, specify use_std_nim to true. On PowerVS VMs we
have /usr/sys/inst.images/installp/ppc in that case we have it as true
use_std_nim: false
powernim_local_path: '/usr/sys/inst.images/installp/ppc'
```

3.3.3) Next update variables related to Oracle user and groups

Below is the example snippet

```
ora_user: oracle
ora_user_uid: 600
ora_group: dba
ora_group_gid: 500
ora_oinstall_group: oinstall
ora_oinstall_group_gid: 501
ora_oper_group: oper
ora_oper_group_gid: 503
ora_home_profile: /home/{{ora_user}}
ora_install_group: oinstall
ora_asmdba_group: asmdba
ora_asmoper_group: asmoper
ora_asmadmin_group: asmadmin
ora_asmdba_group_gid: 504
ora_asmoper_group_gid: 505
ora_asmadmin_group_gid: 506
```

3.3.4) Next update the variable “ignoreprecheck”

ignoreprecheck=false means the oracle installer will do prechecks during installation. If any of the critical checks fails it will halt the execution.

ignoreprecheck=true means the oracle installer will skip the prechecks during installation. If the any of critical checks fails the installation continues.

Eg: If oracle installer fails while doing critical check related to some APAR number and the fix is already exists with different number. Now if you want to skip the critical check failure and continue with execution then set flag “ignoreprecheck=true”.

3.3.5) Next specify the OS paging size in MB, hd6 lv is used for paging by default.

```
os_paging_size: 16384
```

3.3.6) Next update the variables of oracle DB and specify the storage disks for Oracle Homes staging.

Below is the example snippet for creating orcl database and oracle homes on /u01 filesystem.

```
ora_sid: orcl
ora_pwd: Oracle4u
ora_character_set: WE8MSWIN1252

ora_sw_vg: oravg
ora_sw_vg_disk: ['hdisk1','hdisk2']
ora_sw_fs: "/u01"
ora_sw_fs_size: 200G
ru_tgt_path: "{{ora_sw_fs}}/RU19"
ora_inventory: "{{ora_sw_fs}}/oraInventory"
ora_base: "{{ora_sw_fs}}/base"
ora_home: "{{ora_sw_fs}}/product/{{ora_version}}/database"
```

3.3.7) If database on JFS2 set below parameters for creating the volume group and filesystem for datafiles, control file and redo log files staging.

Below is the example snippet

```
ora_data_vg: oradatavg
ora_data_vg_disk: ['hdisk3','hdisk4']
ora_data_fs: /oradata
ora_data_fs_size: 5G
ora_db_fs_list:
- { fs_name: "/datafiles", size: "8G" }
- { fs_name: "/controlfiles", size: "2G" }
- { fs_name: "/redologs", size: "4G", bs: 512 }
```


3.3.8) If Database on ASM set “grid_asm_flag” to true otherwise set to false. Update the variables that are needed for grid install

Below is the example snippet

```
grid_asm_flag: true
grid_ora_home: "{{ora_sw_fs}}/grid/home"

# Grid reponsefile parameter
ora_grid_user: grid
ora_grid_user_uid: 601
asm_sys_password: oracle
asm_diskgroup_name: DATA
asm_diskgroup_redundancy: EXTERNAL
asm_diskgroup_au_size: 4
asm_diskgroup_disk: ['hdisk20','hdisk21']
asm_disk_loc_prefix: "/dev/r"
asm_diskdiscoverystring: "{{asm_disk_loc_prefix}}hdisk*"
asm_asmsnmp_password: oracle4U
```

Things to be taken care

- The disks that are used for oracle installation and oracle ASM disks should be clean – disk headers should not contain old data. To clear pvid for disk use 'chdev -l hdiskX -a pv=clear' and to clear header info use 'dd if=/dev/zero of=/dev/hdiskX bs=1024k count=100' You can check the header information using “lquerypv -h /dev/hdiskX”.
- The rootvg disk should be atleast 30GB, we will be using /tmp for ansible remote location(~8GB) and assumes paging device part of rootvg

Note : When Running playbook on PowerVS AIX VM we noted that by default rootvg (boot) disk size is 20G, Before running the playbook we need to extend the rootvg size by adding new disk. For example “extendvg -f rootvg newhdiskX”

3.4) Based on your environment update resolv.conf and netsvc.conf files at “<collection_dir>/roles/preconfig/files/”

3.5) Create/Update ansible.cfg and inventory files in collections directory. On managed host(AIX) “/tmp” filesystem is used for ansible remote temporary activities. Since we need to stage oracle binary software files, the playbook will automatically set the /tmp filesystem size to 8G. “inventory” file should contain the list on managed hosts (AIX lpars).

```
$ cat ansible.cfg
[defaults]
inventory = ./inventory
interpreter_python = /opt/freeware/bin/python3
remote_user = root
host_key_checking = False
remote_tmp = /tmp/.ansible
[ssh_connection]
ssh_args = -o ForwardAgent=yes -o ControlPersist=30m -o ServerAliveInterval=45 -o
ServerAliveCountMax=10
```

```
$ cat inventory
aixhost
```

4. Execute playbook using below command

If yum/dnf and python is not configured on AIX system, first execute the bootstrap playbook from ibm.power_aix collection, refer to Appendix section for steps.

Note: By default dnf package will be installed at /opt/freeware/bin. If not, use “demo_bootstrap_dnf.yml” from ibm.power_aix collection for configuring dnf and python3

Once yum/dnf and python got configured on managed host then you can run below playbook for installing oracle binary and creating test database

```
$ cat demo_play_aix_oracle.yml
- hosts: all
  gather_facts: yes
  vars_files: vars/oracle_params.yml
  roles:
    - role: preconfig
      tags: preconfig
    - role: oracle_install
      tags: oracle_install
    - role: oracle_createdb
      tags: oracle_createdb

$ ansible-playbook demo_play_aix_oracle.yml
```

You can also run each role separately using ansible tags

To run only preconfig tasks

```
$ ansible-playbook demo_play_aix_oracle.yml --tags "preconfig"
```

If you want to run preconfig and oracle_install tasks

```
$ ansible-playbook demo_play_aix_oracle.yml --tags "preconfig,oracle_install"
```

If you want to skip database creation tasks then you can also try

```
$ ansible-playbook demo_play_aix_oracle.yml --skip-tags="oracle_createdb"
```

If you are installing the grid by default cluvfy prechecks will get executed. If you want to skip cluvfy precheck task you can use the skip-tag

```
$ ansible-playbook demo_play_aix_oracle.yml --skip-tags=runcluvfy
```

Troubleshooting

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

```
$ ansible-playbook demo_play_aix_oracle.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>

Playbook Roles

As discussed earlier this collection has three roles

1) preconfig :

- Expand /var and /opt filesystems
- Running cfgmgr to discover new devices
- Changes /tmp to 12G size, holds ansible temp files
- Changing ulimits for default user to unlimited
- Check rpm Packages and use dnf for installing rpm packages.
- Setting DNS
- Checking /etc/hosts file on managed host and adding entry if needed.
- Changes maxuproc
- Set OS paging size
- Do VG disks and ASM disk validations
- Checking and setting iocp attribute to "available".

- Creating NFS filesystems for filesets.
- Check and Install Filesets.
- Rebooting the lpar if needed

2) oracle_install:

- Creating NFS filesystem from nfshost for Oracle Binaries.
- Detecting Oracle version to install.
- Create Oracle groups and user.
- Creating volume group for ORACLE_HOME.
- Creating and mounting filesystem for ORACLE_HOME.
- Creating Oracle installation directories.
- Check if Oracle home is installed; if not installed, then install.
- If the RU option is selected, extract RU binaries and OPatch from NFS/remote/local.
- If the grid option is selected, install Standalone Grid Software.
- Updating the .profile file with Oracle env details.
- Generating Oracle response file.
- Install Oracle DB/Grid Software with the base version and with the RU version, with the option to include or exclude the ignoreprecheck parameter.
- Apply RU if the Oracle version is greater than 19.17.
- Run root scripts.

3) oracle_createdb:

- Check /etc/oratab file for DB existence
- If grid option selected create database on ASM storage
- For JFS DB, create VG and mount filesystems
- Generate Database creation template file
- Generate database creation script
- Creating database
- For JFS DB, Creating and configuring oracle listener
- Check Oracle PMON background process status
- Add TNS Entry to tnsnames.ora.

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-ee/html-single/ansible_builder_guide/index

```
$ cat execution-environment.yml
---
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 dnf
dependencies:
  galaxy: requirements.yml
```

```
$ cat requirements.yml
---
collections:
  - ibm.power_aix
```

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

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

Displaying the content of container or docker file

```
$ cd context
$ cat Containerfile
ARG EE_BASE_IMAGE="registry.redhat.io/ansible-automation-platform-24/ee-minimal-
rhel8:latest"
ARG PYCMD="/usr/bin/python3"
ARG PKGMGR_PRESERVE_CACHE=""
ARG ANSIBLE_GALAXY_CLI_COLLECTION_OPTS=""
ARG ANSIBLE_GALAXY_CLI_ROLE_OPTS=""
ARG PKGMGR="/usr/bin/microdnf"

# Base build stage
FROM $EE_BASE_IMAGE as base
USER root
ARG EE_BASE_IMAGE
ARG PYCMD
ARG PKGMGR_PRESERVE_CACHE
ARG ANSIBLE_GALAXY_CLI_COLLECTION_OPTS
ARG ANSIBLE_GALAXY_CLI_ROLE_OPTS
ARG PKGMGR

RUN $PYCMD -m ensurepip
COPY _build/scripts/ /output/scripts/
COPY _build/scripts/entrypoint /opt/builder/bin/entrypoint
RUN microdnf install dnf

# Galaxy build stage
FROM base as galaxy
ARG EE_BASE_IMAGE
ARG PYCMD
ARG PKGMGR_PRESERVE_CACHE
ARG ANSIBLE_GALAXY_CLI_COLLECTION_OPTS
ARG ANSIBLE_GALAXY_CLI_ROLE_OPTS
ARG PKGMGR

RUN /output/scripts/check_galaxy
COPY _build /build
WORKDIR /build

RUN ansible-galaxy role install $ANSIBLE_GALAXY_CLI_ROLE_OPTS -r requirements.yml --roles-
path "/usr/share/ansible/roles"
RUN ANSIBLE_GALAXY_DISABLE_GPG_VERIFY=1 ansible-galaxy collection install
$ANSIBLE_GALAXY_CLI_COLLECTION_OPTS -r requirements.yml --collections-path
"/usr/share/ansible/collections"

# Builder build stage
FROM base as builder
WORKDIR /build
ARG EE_BASE_IMAGE
ARG PYCMD
ARG PKGMGR_PRESERVE_CACHE
ARG ANSIBLE_GALAXY_CLI_COLLECTION_OPTS
```

```

ARG ANSIBLE_GALAXY_CLI_ROLE_OPTS
ARG PKGMGR

RUN $PYCMD -m pip install --no-cache-dir bindep pyyaml requirements-parser

COPY --from=galaxy /usr/share/ansible /usr/share/ansible

RUN $PYCMD /output/scripts/introspect.py introspect --sanitize --write-
bindep=/tmp/src/bindep.txt --write-pip=/tmp/src/requirements.txt
RUN /output/scripts/assemble

# Final build stage
FROM base as final
ARG EE_BASE_IMAGE
ARG PYCMD
ARG PKGMGR_PRESERVE_CACHE
ARG ANSIBLE_GALAXY_CLI_COLLECTION_OPTS
ARG ANSIBLE_GALAXY_CLI_ROLE_OPTS
ARG PKGMGR

RUN /output/scripts/check_ansible $PYCMD

COPY --from=galaxy /usr/share/ansible /usr/share/ansible

COPY --from=builder /output/ /output/
RUN /output/scripts/install-from-bindep && rm -rf /output/wheels
RUN chmod ug+rw /etc/passwd
RUN mkdir -p /runner && chgrp 0 /runner && chmod -R ug+rw /runner
WORKDIR /runner
RUN $PYCMD -m pip install --no-cache-dir 'dumb-init==1.2.5'
RUN rm -rf /output
LABEL ansible-execution-environment=true
USER 1000
ENTRYPOINT ["/opt/builder/bin/entrypoint", "dumb-init"]
CMD ["bash"]

```

\$podman images

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
localhost/oracle_aix_ee	latest	a7d4b0992ea4	3 minutes ago	475 MB
registry.redhat.io/ansible-automation-platform-24/ee-supported-rhel8	latest	b2d26de2d8de	9 months ago	1.79 GB
registry.redhat.io/ansible-automation-platform-24/ee-minimal-rhel8	latest	c239714e9480	9 months ago	380 MB
quay.io/ansible/ansible-runner	latest	bec0dc171168	23 months ago	816 MB

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

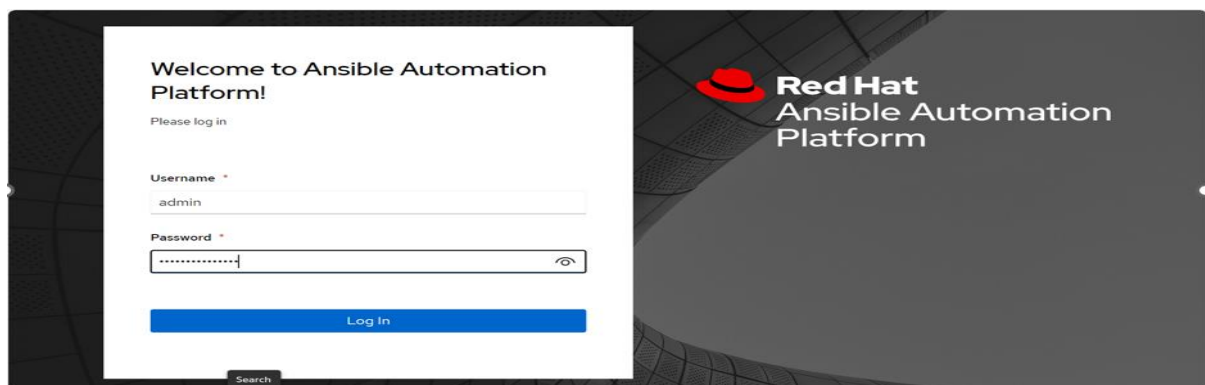
Go to power_aix_oracle collection and create ansible-navigator.yaml file

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

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

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

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



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

Red Hat Ansible Automation Platform

Execution Environments

Create new execution environment

Name *

Image * ⓘ

Pull

Description

Organization

Registry credential ⓘ

Save Cancel

Leave this field blank to make the execution environment globally available.

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

Red Hat Ansible Automation Platform

Execution Environments

Create new execution environment

Name *

Image * ⓘ

Pull

Description

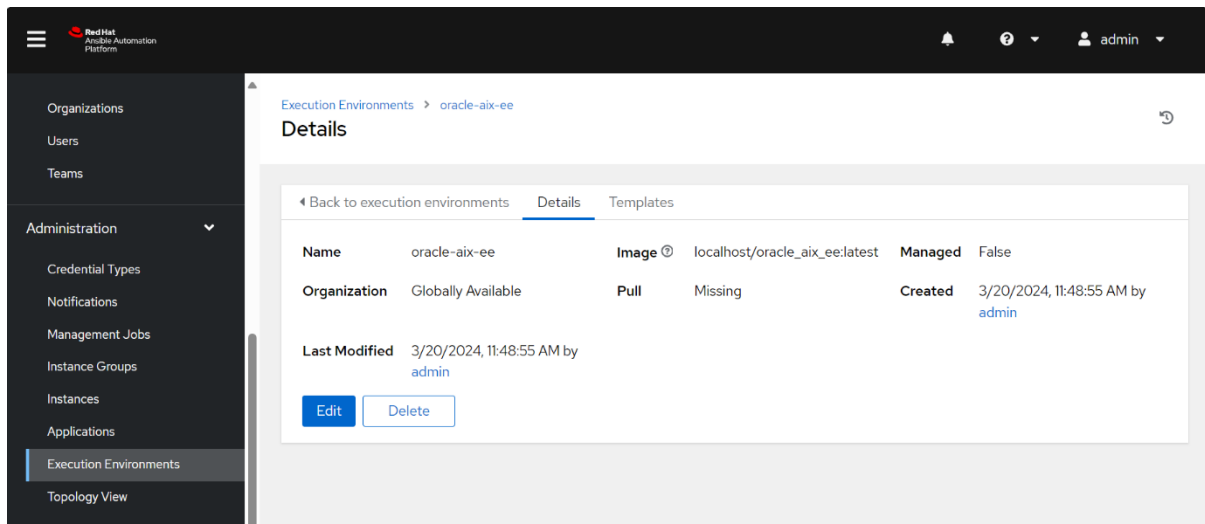
Organization

Registry credential ⓘ

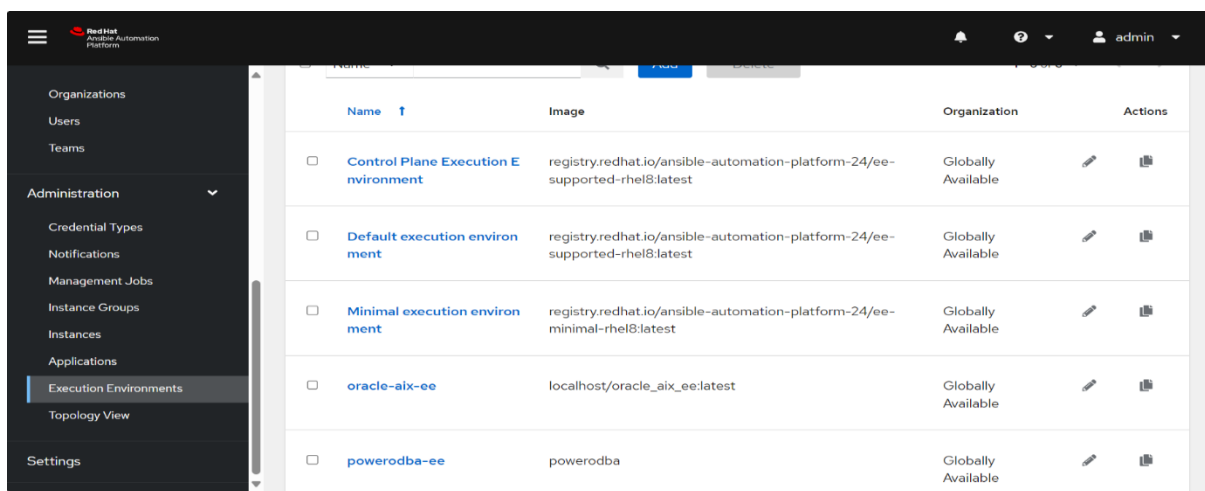
Save Cancel

Leave this field blank to make the execution environment globally available.

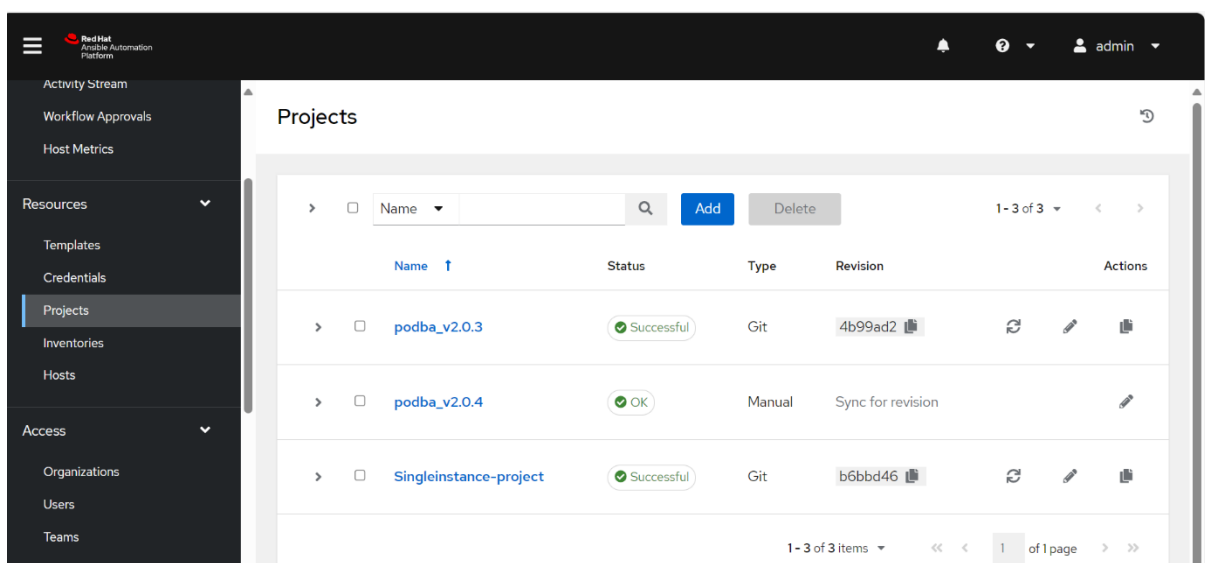
Once saved, we can see the following details to crosscheck.



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: Manual
- Project Base Path: /var/lib/awx/projects [where the collection code is placed]
- Playbook Directory: [Specify the directory where your playbook is located]

Once provided the details, please click on “Save”

The screenshot shows the 'Create New Project' form in the Red Hat Ansible Automation Platform. The form is divided into several sections: 'Name', 'Description', 'Organization', 'Execution Environment', 'Source Control Type', 'Content Signature Validation Credential', 'Type Details', 'Project Base Path', and 'Playbook Directory'. The 'Name' field is filled with 'oracle-aix-test-si', 'Description' with 'oracle', 'Organization' with 'Default', 'Execution Environment' with 'oracle-aix-ee', 'Source Control Type' with 'Manual', 'Project Base Path' with '/var/lib/awx/projects', and 'Playbook Directory' with 'ansible-power-aix-oracle'. The 'Save' button is highlighted in blue.

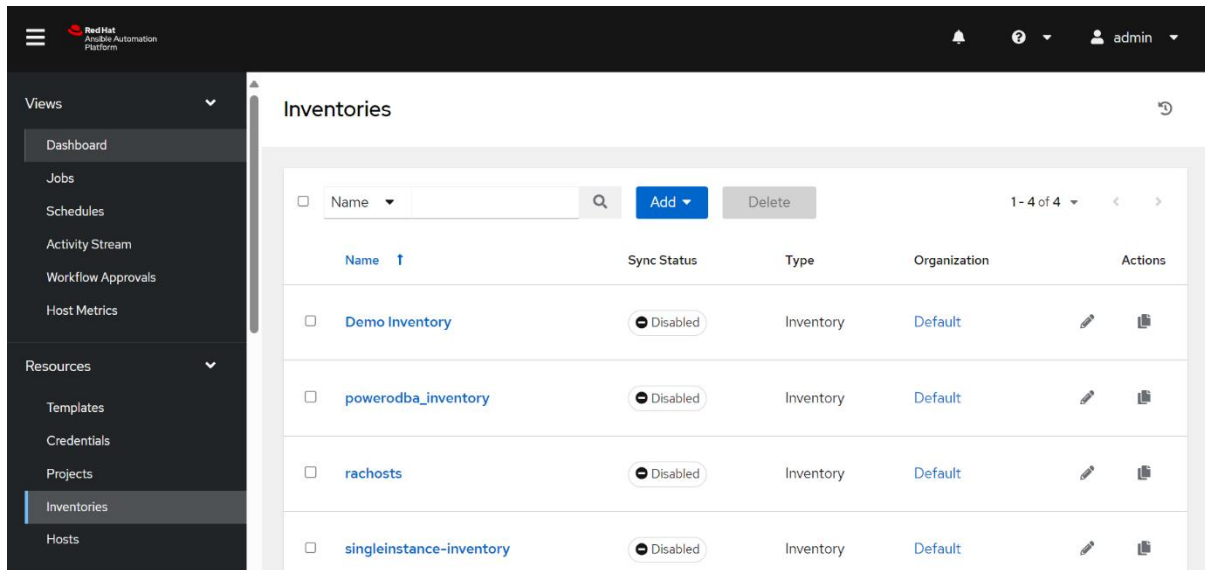
Once saved, we can see the following details to crosscheck.

The screenshot shows the 'Details' page for the project 'oracle-aix-test-si'. The page has a navigation bar with 'Back to Projects', 'Details', 'Access', 'Job Templates', and 'Notifications'. The 'Details' tab is selected. The table below shows the project details:

Name	oracle-aix-test-si	Description	oracle	Organization	Default
Source Control Type	Manual	Source Control Revision	Sync for revision	Cache Timeout	0 Seconds
Default Execution Environment	oracle-aix-ee	Project Base Path	/var/lib/awx/projects	Playbook Directory	ansible-power-aix-oracle
Created	3/20/2024, 11:54:15 AM by admin	Last Modified	3/20/2024, 11:54:15 AM by admin		

At the bottom of the table, there are 'Edit' and 'Delete' buttons.

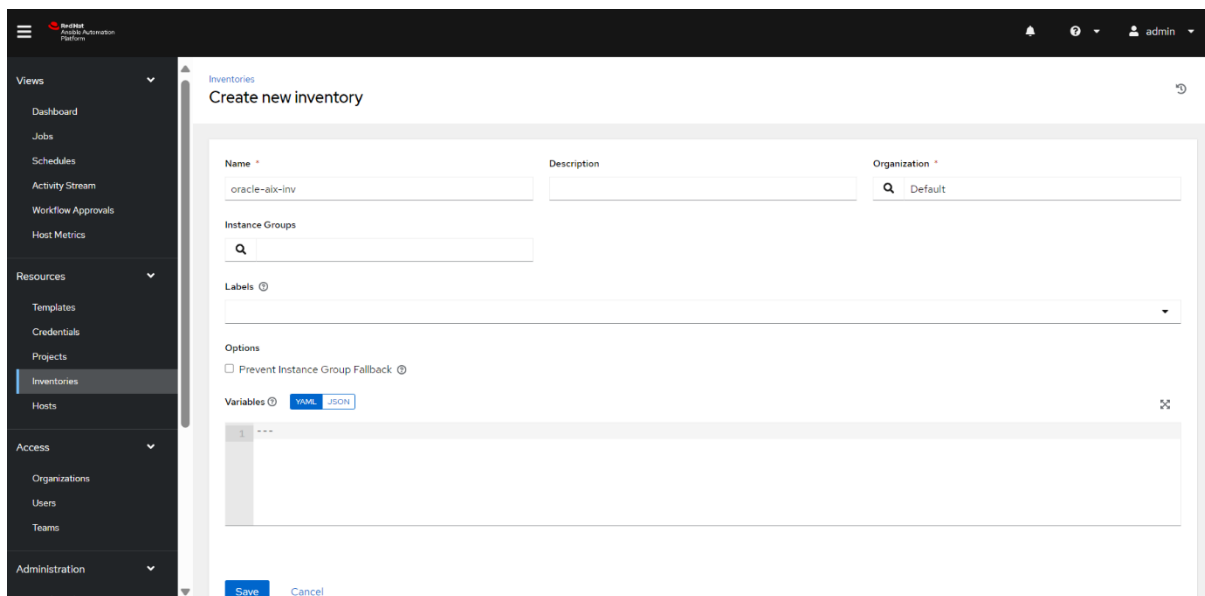
To create Inventories, click on the "Inventories" option under the Resources dropdown and then click on the "Add" button to create Inventories.



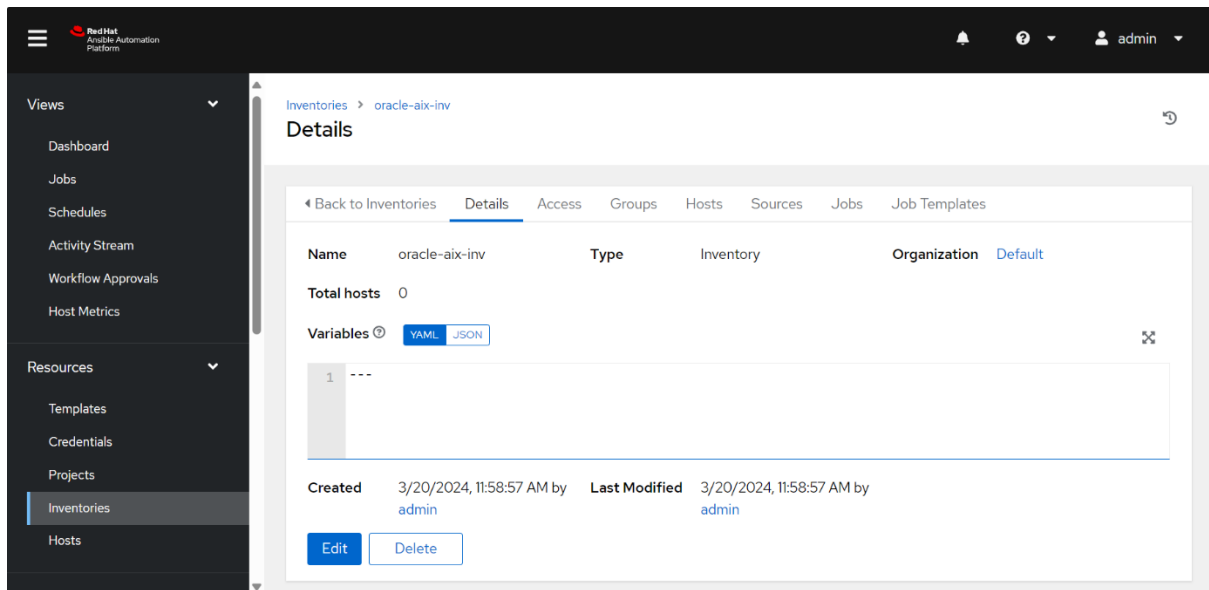
Please provide the following details:

- Name: [Enter the name of your project]
- Description: [Enter a brief description of your project]
- Organization: [Select the organization for your project]

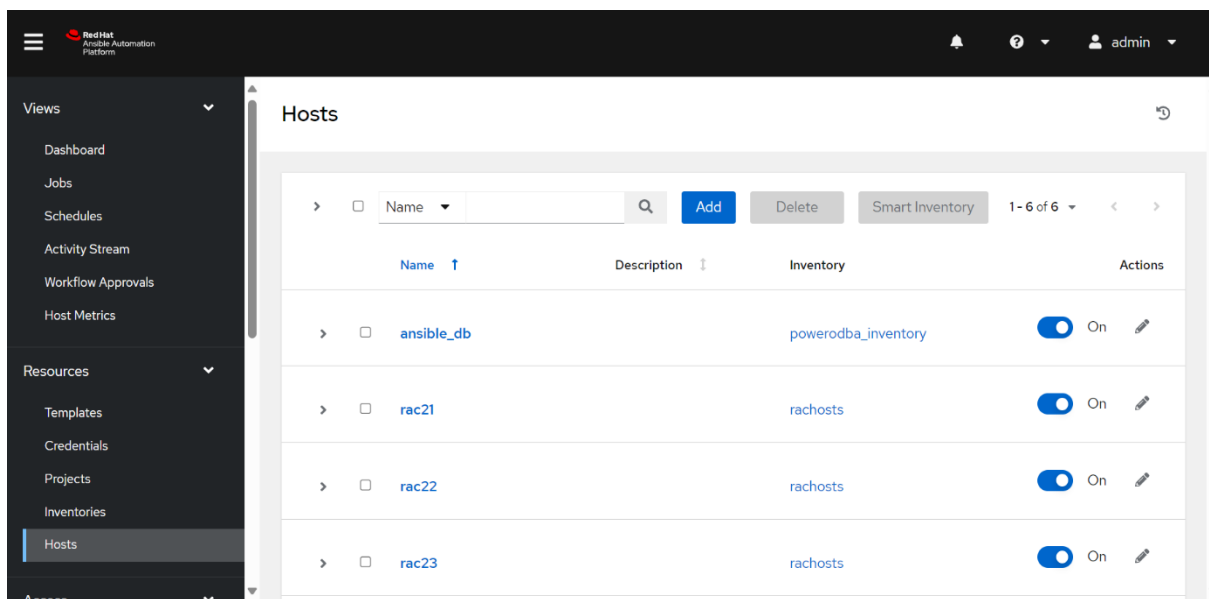
After providing the required information, click on "Save".



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



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



Please provide the following information:

- Name: [Enter the name of your project]
- Description: [Enter a brief description of your project]
- Inventory: [Select the inventory that was earlier created]
- Variables: ansible-host

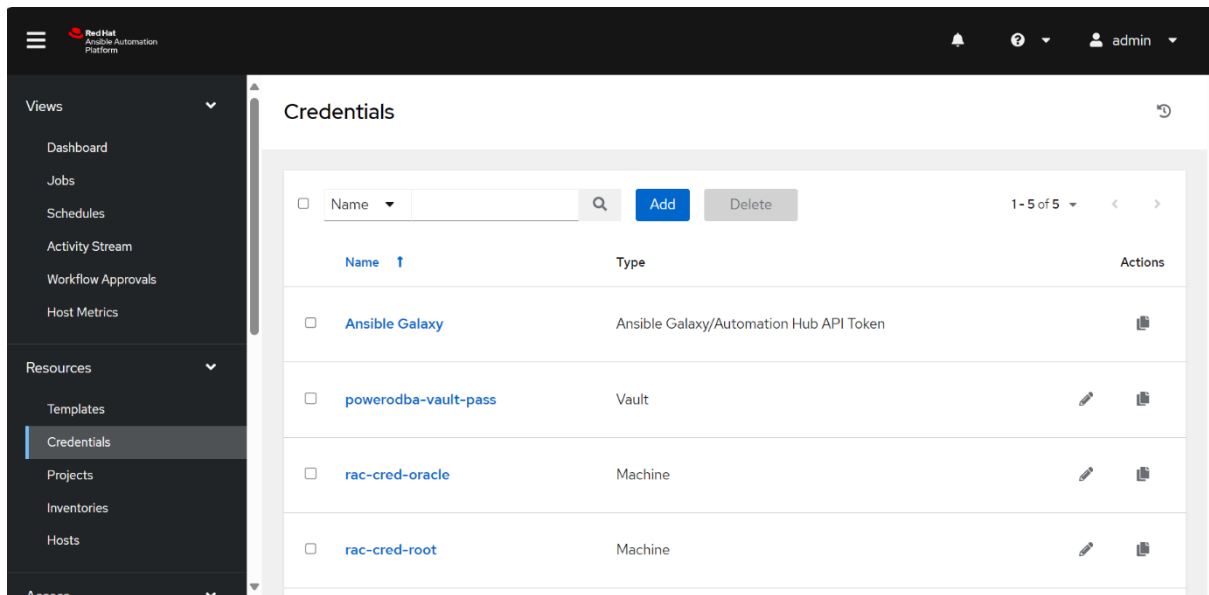
After providing the required information, click on "Save".

The screenshot shows the 'Create New Host' form in the Red Hat Ansible Automation Platform. The left sidebar contains a 'Views' menu with options like Dashboard, Jobs, Schedules, Activity Stream, Workflow Approvals, and Host Metrics. Below it is a 'Resources' menu with options like Templates, Credentials, Projects, Inventories, and Hosts. The main content area is titled 'Create New Host' and contains a form with the following fields: 'Name' (oracle-aix-host), 'Description' (empty), and 'Inventory' (oracle-aix-inv). Below these fields is a 'Variables' section with a 'YAML' tab selected. The variables are defined in a list: 1. ---, 2. ansible-host: [redacted]. At the bottom of the form are 'Save' and 'Cancel' buttons.

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

The screenshot shows the 'Details' page for the host 'oracle-aix-host' in the Red Hat Ansible Automation Platform. The left sidebar is the same as in the previous screenshot. The main content area is titled 'Details' and contains a 'Back to Hosts' link and tabs for 'Details', 'Facts', 'Groups', and 'Jobs'. The 'Details' tab is selected. The page shows the following information: 'Name' (oracle-aix-host), 'Inventory' (oracle-aix-inv), 'Created' (3/20/2024, 12:04:50 PM by admin), and 'Last Modified' (3/20/2024, 12:04:50 PM by admin). Below this is a 'Variables' section with a 'YAML' tab selected. The variables are defined in a list: 1. ---, 2. ansible-host: [redacted]. At the bottom of the page are 'Edit' and 'Delete' buttons.

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



Please provide the following information:

Name: [Enter the name of your credential]

Description: [Enter a brief description of your credential]

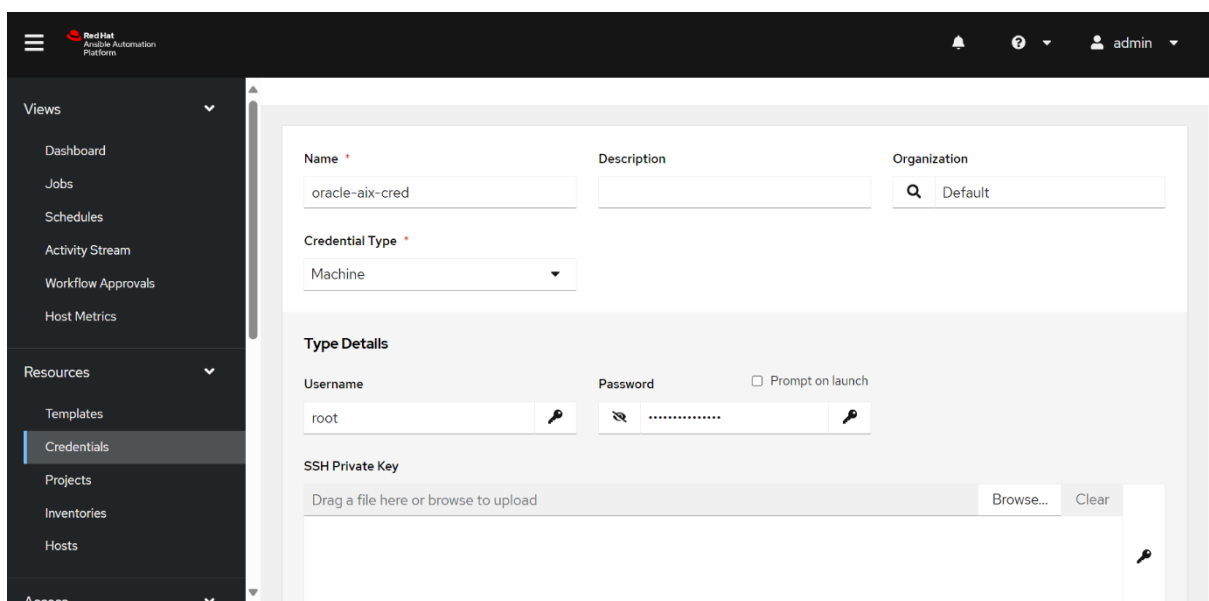
Organization: [Select the organization for your credential]

Credential Type: Machine

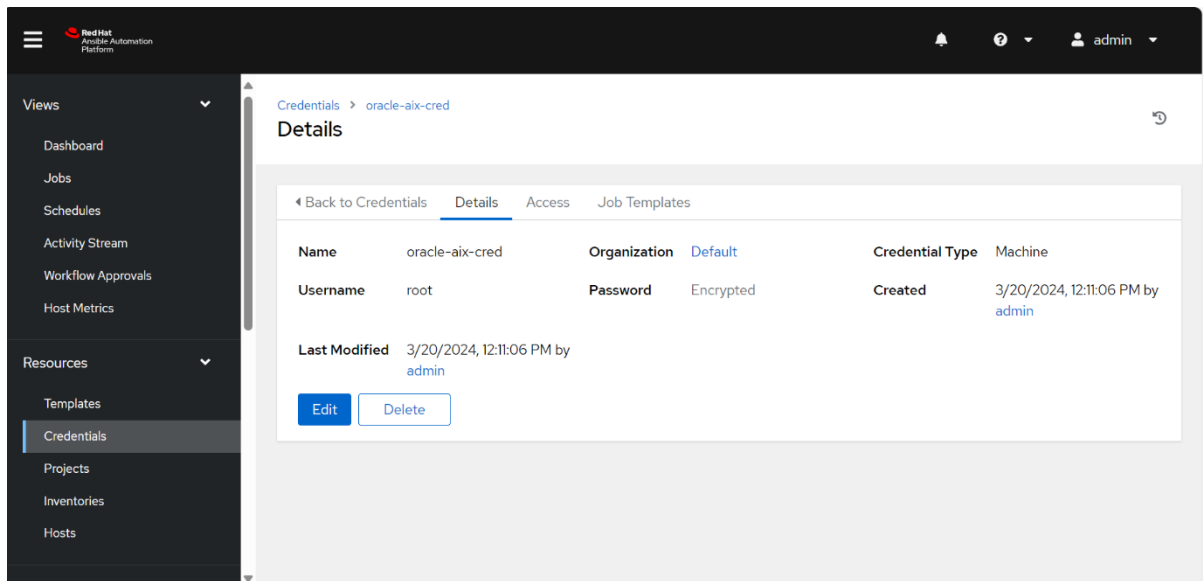
Username: [Enter the username for the machine]

Password: [Enter the password for the machine]

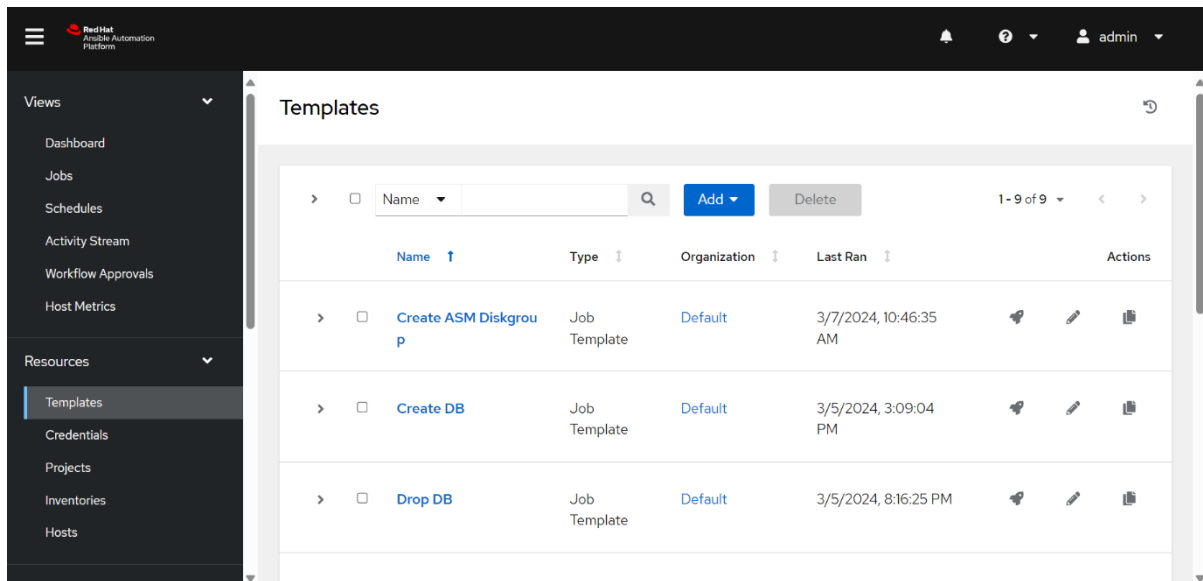
After providing the required information, click on "Save".



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:

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]

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

Create New Job Template

Name * oracle-aiX-template **Description**

Job Type * Run ☐ Prompt on launch

Inventory * ☐ Prompt on launch **Project *** oracle-aiX-test-si

Execution Environment ☐ Prompt on launch
oracle-aiX-ee

Playbook * demo_play_aiX_oracle.yml

Credentials ☐ Prompt on launch
SSH: oracle-aiX-cred

Labels ☐ Prompt on launch

After providing the required information, proceed with executing the job by clicking on the "Launch" button.

Details

Back to Templates Details Access Notifications Schedules Jobs Survey

Name	oracle-aiX-template	Job Type	run	Organization	Default
Inventory	oracle-aiX-inv	Project	oracle-aiX-test-si	Execution Environment	oracle-aiX-ee
Playbook	demo_play_aiX_oracle.yml	Forks	0	Verbosity	0 (Normal)
Timeout	0	Show Changes	Off	Job Slicing	1
Created	3/20/2024, 12:16:13 PM by admin		Last Modified	3/20/2024, 12:17:51 PM by admin	
Credentials	SSH: oracle-aiX-cred				
Job Tags	oracle_install				
Skip Tags	randomify				
Variables	vars, json				

[Edit](#) [Launch](#) [Delete](#)

View the playbook status from stdout console

Output

Back to Jobs Details Output

oracle-aiX-template Elapsed 00:00:00

Stdout Unfollow

```

3
4 TASK [Gathering Facts] ***** 00:36:17
5 ok: [169.63.62.166]
6
7 TASK [preconfig : Check current /var size] ***** 00:36:30
8 changed: [169.63.62.166]
9
10 TASK [preconfig : Expand /var target directory +500M] ***** 00:36:31
11 skipping: [169.63.62.166]
12
13 TASK [preconfig : Check current /opt size] ***** 00:36:31

```

PowerVC – Infrastructure Provisioning

IBM Power Virtualization Center (PowerVC) provides simplified management of IBM AIX, IBM i and Linux virtual machines (VMs) running on IBM Power. It is built on OpenStack to provide private cloud capabilities across your IBM Power environment. IBM PowerVC capabilities include being able to create and destroy VMs, networks, network interfaces, storage volumes and images. It also has the ability to perform tasks against the VMs such as stop, start, resize, migrate, clone, create and restore snapshots, and attach storage volumes.

Inside “power_aix_oracle” collection we add new role “powervc_create_aixvm” for creating AIX operating system VM which is needed for Oracle deployment. Once VM is created the required number of disks are created and attached to the VM for Oracle Deployment.

a) Preparing the ansible controller

Install the OpenStack Cloud collection from ansible galaxy

```
ansible-galaxy collection install openstack.cloud
```

For Ansible to run the OpenStack Cloud modules, you must first install the OpenStack SDK on your Ansible controller.

```
pip3.9 install openstacksdk
```

Before Ansible can use the OpenStack cloud modules it needs to authenticate with the IBM PowerVC server.

You can copy the /opt/ibm/powervc/powervcrc file and /etc/pki/tls/certs/powervc.crt file to collection directory or any other directory on ansible controller node. Update the certificate powervc.crt file location in powervcrc file. Before running the playbook run “source powervcrc” which will set the powervc environment variables.

Another method to authenticate from the command line is create a ‘clouds.yml’ file which contains the information about the cloud environments that Ansible needs to connect to IBM PowerVC server. The OpenStack modules will look for the clouds.yml file in the following directories:

- current directory
- ~/.config/openstack
- /etc/openstack

Sample clouds.yml file

```

powervc_cloud:
  auth:
    auth_url: https://x.x.x.x:5000/v3/
    project_name: ibm-default
    project_domain_name: Default
    user_domain_name: Default
    username: <powervc_userid>
    password: <powervc_userid_password>
    region_name: RegionOne
    cacert: "<path>/powervc.crt"

```

Another way to do the authentication is to have powervc environment variables defining them in playbook itself, For example

```

---
- name: "Play1: Creating PowerVC VM"
  hosts: localhost
  gather_facts: yes
  environment:
    OS_AUTH_URL: https://x.x.x.x:5000/v3/
    OS_USERNAME: <powervc_userid>
    OS_PASSWORD: <powervc_userid_password>
    OS_CACERT: "<path>/powervc.crt"
    OS_REGION_NAME: RegionOne
    OS_USER_DOMAIN_NAME: Default
    OS_PROJECT_DOMAIN_NAME: Default
    OS_PROJECT_NAME: ibm-default
...

```

b) Create PowerVC image (optional)

Using powervc capture feature we can create new images from existing VMs. This step is optional. Using the newly created Image will skip some of the tasks during the execution.

- 1) Create VM having AIX 72 or AIX 73 latest versions. Ensure rootvg disk of 40G
- 2) Install the AIX fileset xlftrte.aix61
- 3) Install the unzip rpm package
- 4) If dnf and python3 is not present run "demo_bootstrap_dnf.yml" playbook from ibm.power_aix collection for configuring dnf and python3. If required rpms were installed manually, dnf config is optional. By default python3 will be available in AIX 73 standard images.
- 5) Capture the VM and create New image

Note : If your increasing the rootvg disk size, make sure you run "chlv -x 1024 hd6" and "chvg -g rootvg", so that rootvg Total PPs get updated

c) Update the variable file vars/powervc_oracle_params.yml

The Variable file "powervc_oracle_params.yml" contains both powerVC variables and Oracle variables. Below are the powerVC specific variables

```
vm_name: "<Provide vm name>"
cimage_id: "Provide PowerVC image id"
vm_profile: "Provide flavor type from powerVC"
vm_network: "Provide Network name from powerVC"
orasw_vg_disk_size: <Provide the disk size for staging oracle home, provide min of 100G>
data_disk_count: <Provide the data disks count>
data_disk_size: <Provide size of a data disk>
data_disk_prefix: "{{ vm_name }}_data"
powervc_storage_template: "Name of the storage template in PowerVC"
```

Below is the sample values for powerVC specific variables

```
vm_name: "oravm"
cimage_id: "9437d7e3-14de-4865-a13b-12479fa05909"
vm_profile: "medium"
vm_network: "Network129"
orasw_vg_disk_size: 100
data_disk_count: 3
data_disk_size: 20
data_disk_prefix: "{{ vm_name }}_data"
data_disks: "{%- for n in range(2, data_disk_count + 2, 1 ) -%}
{%- if n == data_disk_count+1 -%} hdisk{{ n }} {%- else -%}
hdisk{{ n }}, {%- endif -%} {%- endfor -%}"
powervc_storage_template: "FS5000 base template"
```

For more details regarding the updating of oracle specific variables refer Step 3 of page4

d) Set the PowerVC environment variables

In step a – “Preparing ansible control” for PowerVC readiness we have mentioned different methods of doing the Authentication. When using powervcrc file, set the powervc environment variables

```
$ source powervcrc
```

e) Run the Playbook

Below is the content of playbook, it contains two plays. Play1 – Creates VM on PowerVC for Oracle installation and Play2- Does the Oracle Deployment

```
$ cat powervc_oracle_play.yml
---
- name: "Play1: Creating PowerVC VM"
  hosts: localhost
  gather_facts: yes
  vars_files: vars/powervc_oracle_params.yml
  vars:
    ansible_python_interpreter: /usr/bin/python
  roles:
    - role: powervc_create_aixvm
      tags: powervc_create_aixvm

- name: "Play2: Deploying Oracle SI DB"
  hosts: "{{ vm_name }}"
  gather_facts: yes
  vars_files: vars/powervc_oracle_params.yml
  vars:
    ansible_python_interpreter: /opt/freeware/bin/python3
  roles:
    - role: preconfig
      tags: preconfig
    - role: oracle_install
      tags: oracle_install
    - role: oracle_createdb
      tags: oracle_createdb
```

Playbook can be executed using

```
$ ansible-playbook powervc_oracle_play.yml | tee test_powervc_oracle.out

$ nohup ansible-playbook powervc_oracle_play.yml > test_powervc_oracle.out 2>&1 &
```

Appendix

In this section we will discuss different ways of installing dnf/python on AIX

AIX7.3/7.2 supports python3 and dnf – configured at /opt/freeware/bin/ directory

if not, The following procedures will help you install the components.

Option A, Using the ansible playbook collection: ansible-power-aiX to install missing components

- Install and configure yum/dnf, python on the managed host. You can do this using ansible power-aiX collection. Power-aiX collection can also perform other AIX admin tasks too.
<https://ibm.github.io/ansible-power-aiX/index.html>
https://galaxy.ansible.com/ui/repo/published/ibm/power_aiX
<https://github.com/IBM/ansible-power-aiX>

Install

Ansible Content for IBM Power Systems – AIX provides a collection of content used to manage and deploy Power Systems AIX.

ibm power aix infrastructure

License GPL-3.0-only

Installation `ansible-galaxy collection install ibm.power_aiX`

Note: Installing collections with ansible-galaxy is only supported in ansible-core>=2.13.9

Download [Download tarball](#)

Requires Ansible >=2.14.0

Download the power-aiX tarball from ansible-galaxy for offline use or you can run the installation command:

```
➤ ansible-galaxy collection install ibm.power_aiX
```

The above command will install power_aiX 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

- a) Prepare demo_dnf_bootstrap.yml using
“~/.ansible/collections/ansible_collections/ibm/power_aiX/playbooks/
demo_dnf_bootstrap.yml”

```
$ cat demo_dnf_bootstrap.yml
---
---
- name: "Bootstrap Yum on AIX"
  hosts: all
  gather_facts: no
  user: root
  collections:
    - ibm.power_aix
  tasks:

# CHECK for Yum on inventory host
- import_role:
    name: power_aix_bootstrap
  vars:
    pkgtype: "dnf"
    download_dir: "~"
    target_dir: "/tmp/.ansible.cpubdir"
```

- b) Prepare ansible.cfg, inventory files for playbook execution. "aixhost" is AIX managed host mentioned in inventory file. Update the "roles_path" to power-aix collection roles directory.

```
$ cat ansible.cfg

[defaults]
inventory = ./inventory
interpreter_python = /usr/bin/python3
remote_user = root
host_key_checking = False
remote_tmp = /tmp/.ansible
roles_path =
/home/ansible/.ansible/collections/ansible_collections/ibm/power_aix /roles

$ cat inventory

aixhost
```

- c) Executing bootstrap playbook

```
$ ansible-playbook demo_dnf_bootstrap.yml
```

Option B, installing them manually onto the managed host.

<https://public.dhe.ibm.com/aix/freeSoftware/aixtoolbox/ezinstall/ppc/>

Download `dnf_bundle_aix_XX.tar` bundle and execute the setup script