Contents

Description

This Ansible collection can be used to install Oracle database 19c Single Instance Database on AIX operating system, with either JFS or Oracle ASM as storage. This playbook was tested on AIX 7.3 and PowerVS AIX partition. This is also tested using Ansible Automation Platform 2. The steps are provided in the later sections of this document.

Versions

1.3.0

- oracle_install role is separated into two roles oracle_install_gi & oracle_install_db
- oracle_create_gold_image roles is introduced to create Gold Images for GI & DB from an existing Single Instance setup
- Supports Gold Image based Single instance GI & DB deployments

1.2.2

- Replaced dnf module with shell command
- Added nameserver file backup task
- Fixed the issue with the latest Opatch v44 while applying patch
- Added idempotency for patch task

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=runcluvfy
- Role separation for Grid and DB homes
- RU apply option during the installation, supports 19.4 or later

1.1.1

- Supports Oracle deployments on AIX 7.3 and PowerVS using AAP2 Execution Environment.

1.1.0

- Supports 19c DB install on JFS2 and ASM

1.0.0

- Supports 19c DB install on JFS2 only

Getting Started with Ansible

This collection assumes the following that:

- The user is familiar with Ansible and should have basic knowledge on YAML, for the purpose of running this playbook
- The user is familiar with Oracle Database Configuration
- The user is familiar with the AIX Operating system
- 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).
- The playbook assumes a **New AIX LPAR** for execution
- The targeted AIX LPAR for installing the Oracle single instance database will be referred to within the rest of the document as the 'host' or 'managed host'.
- The version of Oracle Standalone Database tested is 19.3.0.0 + RU (Optional)
- Dependents on ibm.power_aix collection.

To get started with Ansible refer to

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

To get started with Oracle Database on AIX refer to

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 to

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

This collection contains the following roles:

- ➢ **preconfig**: This role will perform AIX configuration tasks that are needed for oracle installation
- ➢ **oracle_install_gi**: This role performs oracle GI binary installation
- ➢ **oracle_install_db:** This role performs oracle RDBMS binary installation

Note: For customized options for creating the database like a Multitenant architecture, number of pluggable databases, memory allocation etc., please use ibm.power_aix_oracle_dba collection. Refer: https://galaxy.ansible.com/ui/repo/published/ibm/power_aix_oracle_dba

- ➢ **oracle_createdb**: This role creates test database “orcl” using dbca utility
- ➢ **powervc_create_aixvm:** This role creates the AIX lpar in a PowerVC environment.
- ➢ **oracle_create_gold_image:** This role creates the Golden images for Single instance GI & RDBMS software from an existing setup.
- ➢ **oracle_deinstall:** This role deinstalls the Oracle GI & RDBMS software.

System Configuration

We are going to show how to deploy Single Instance Oracle Database on ASM in AIX 7.3.

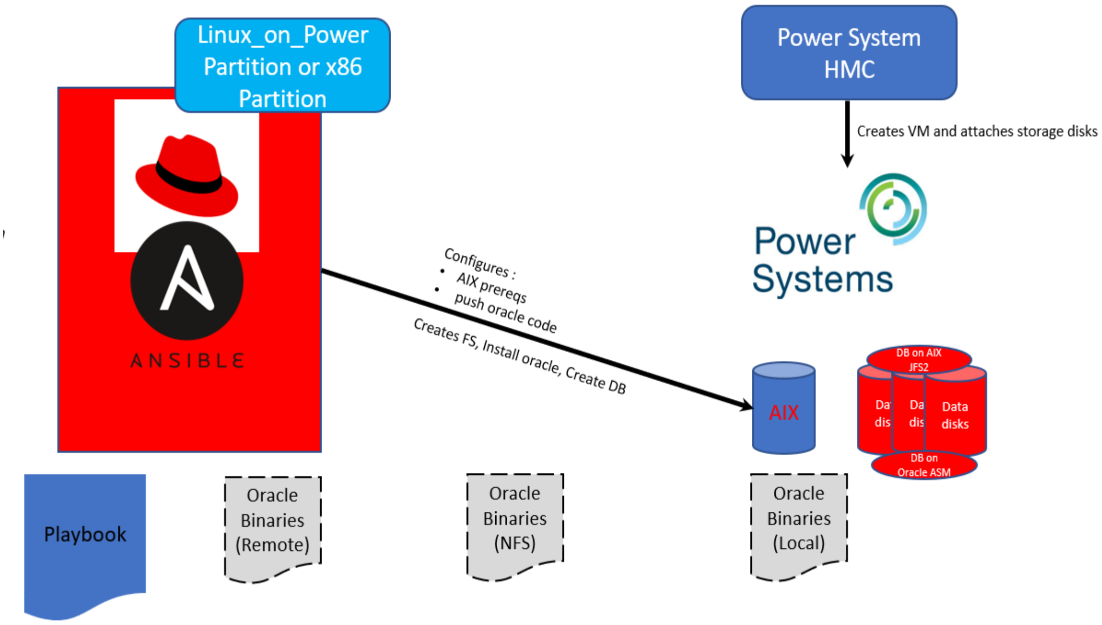


Figure. System Topology

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

We have used two servers, one Redhat 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	: 19.26.0.0
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

Steps

The following section provides the detailed steps for preparing and running the playbook.

Note: Throughout this section, we will be using the hostname "aixhost" for the target AIX lpar.

1. **Install the Ansible Controller on a preferred operating system.**
   Refer: https://docs.ansible.com/ansible/latest/installation_guide/intro_installation.html

A user named "ansible" was created on the Linux On Power server and /home/ansible was chosen as the working directory.

2. **Setup ssh user equivalence with the managed host (AIX) server**

If this is the first time using ssh, then the ssh keys will not be present. To check, change directory to ~/.ssh and check if id_rsa file exists. If not, the ssh keys must be created as follows.

```
➢ ssh-keygen
```

Copy the keys to the managed host using this command and provide the password when prompted.

```
➢ ssh-copy-id root@aixhost
```

Verify the ssh connectivity, it should login without asking for the password

```
➢ ssh root@aixhost
```

3. **Preparing to run the oracle playbook**

   3.1. This ansible collection power_aix_oracle leverages the ibm.power_aix collection modules like the filesystem, devices, lvg and mount. Install the latest version of ibm.power_aix collection from galaxy using the below command.

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

   The power_aix_oracle collection is available in the Ansible Galaxy and Github.

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

   3.2. Download the power_aix_oracle tarball from https://galaxy.ansible.com/ui/repo/published/ibm/power_aix_oracle and extract it to a directory for offline use **or** it can be installed using the below command

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

The above command will install power_aix_oracle collection in ~/.ansible/collections path.

For more information regarding the ansible collections refer the below link

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

3.3. Download the Oracle 19c software from OTN or Oracle e-delivery site

https://edelivery.oracle.com/osdc/faces/SoftwareDelivery
https://www.oracle.com/database/technologies/oracle19c-aix-193000-downloads.html

In case of Release Update (RU) patch apply along with the installation, download the respective patch and the latest Opatch from Oracle support site https://support.oracle.com

3.4. Update the variables in oracle_params.yml file. The sample files are provided in <collection dir>/playbooks/vars/sample/oracle_params_asm.yml & <collection dir>/playbooks/vars/sample/oracle_params_jfs.yml.

In this example, we're going to create a database on ASM. Hence, the variable file oracle_params_asm.yml will be used. In case, the DB should be created on a file system, please use oracle_params_jfs.yml variable file.

3.4.1. Provide the ansible work directory on target AIX system. Do not remove the files under this directory until the playbook execution is complete.

```
work_dir: /tmp/si_ansible
```

3.4.2. Provide the oracle version which is going to be installed. 19c is the allowed value for this parameter.

```
ora_version: 19c
```

3.4.3. Update resolv.conf file: Setting this to true will backup /etc/resolv.conf file and copies the file from "roles/preconfig/files/resolv.conf" to /etc.
Make sure to update the nameserver details in roles/precofig/files/resolv.conf before proceeding.
If set to false, no changes will be done to the /etc/resolv.conf file

```
modify_nameserv: false
```

3.4.4. Update the RPM packages list which are required. In the following box, the unzip and bash utilities will be installed.

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

3.4.5. Provide the nfs details for the installation of filesets. This is optional and the values are blank by default.

```
nfsfilesetMountPoint: /filesets
nfsfilesetExportDir: /filesets
nfsfilesetServerIP: 192.168.1.1
```

3.4.6. If nim filesystem is mounted on the target host, specify use_std_nim: true. On PowerVS VMs, the standard path is /usr/sys/inst.images/installp/ppc

```
use_std_nim: false
powernim_local_path: '/usr/sys/inst.images/installp/ppc'
```

3.4.7. Specify OS paging (swap) size in MB. Oracle recommends 16GB.

```
os_paging_size: 16384
```

3.4.8. Provide the required OS Groups and IDs for the GI & RDBMS software owners.

```
ora_oinstall_group: oinstall
ora_oinstall_group_gid: 501
ora_group: dba
ora_group_gid: 500
ora_oper_group: oper
ora_oper_group_gid: 503
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
```

3.4.9. Provide the Grid s/w owner name and group ID

```
ora_grid_user: grid
ora_grid_user_uid: 601
```

3.4.10. Provide the RDBMS s/w owner name and group ID

```
ora_user: oracle
ora_user_uid: 600
```

3.4.11. Provide the details of the disk and mount point path for Grid & Database s/w installation.

```
ora_sw_vg: orafsvg            # preferred volume group name
ora_sw_vg_disk:  ['hdisk1'] # Physical disk names
ora_sw_fs: "/u01"             # Preferred file system name
ora_sw_fs_size: 99G            # Required size of the file system
```

3.4.12. **Software staging path:** It can be nfs|local|remote. Choose only one.

3.4.12.1. When the software zip files are on an NFS server, provide the value for **ora_binary_location as nfs.**

```
ora_binary_location: nfs
```

Provide the nfs host, nfs mount and the nfs filesystem details as shown below.

```
ora_nfs_host: 192.168.1.1
ora_nfs_device: /repos
ora_nfs_filesystem: /binora
database_sw: /binora/images/oracle/19c/V982583-01_193000_db.zip
grid_sw: /binora/images/oracle/19c/V982588-01_193000_grid.zip
ru_file: /binora/images/oracle/19c/RU19.26/p37257886_190000_AIX64-5L_GI_RU19.26.zip
opatch_file: /binora/images/oracle/opatch/12.2.0.1.44/p6880880_121010_AIX64-5L.zip
```

3.4.12.2. When the software zip files are in the target host (AIX) lpar, provide the value for **ora_binary_location as local**

```
ora_binary_location: local
```

Provide the full path of the software zip files which are in the AIX lpar

```
database_sw: /stage/19c/V982583-01_193000_db.zip
grid_sw: /stage/19c/V982588-01_193000_grid.zip
ru_file: /stage/RU/p37257886_190000_AIX64-5L_GI_RU19.26.zip
opatch_file: /stage/OPatch/12.2.0.1.44/p6880880_121010_AIX64-5L.zip
```

3.4.12.3. When the software zip files are in the ansible controller, provide the value for **ora_binary_location as remote**

```
        ora_binary_location: remote
```

Provide the full path of the software zip files which are in the ansible controller

```
database_sw: /backup/19c/V982583-01_193000_db.zip
grid_sw: /backup/19c/V982588-01_193000_grid.zip
ru_file: /backup/RU/p37257886_190000_AIX64-5L_GI_RU19.26.zip
opatch_file: /backup/OPatch/12.2.0.1.44/p6880880_121010_AIX64-5L.zip
```

3.4.12.4. **Gold Image:** The deployment can be done using gold images as well. Provide the Gold image zip filename along with the path to the grid_sw & database_sw variables.

3.4.13. To apply the Release update (RU) patch along with the installation, set “apply_ru: True”. In case of a **gold image**, set this variable to False “apply_ru: False”.

```
apply_ru: True
```

3.4.14. Provide the path for the Oracle Base & Inventory directories.

```
ora_base: "{{ora_sw_fs}}/base"
ora_inventory: "{{ora_sw_fs}}/oraInventory"
```

3.4.15. Provide the path where the Grid Infrastructure for Standalone needs to be installed.

```
grid_ora_home: "{{ ora_sw_fs }}/19c/grid/home"
```

3.4.16. Provide the path where the Release update (RU) patches should be extracted on the target AIX lpar. This is required only when apply_ru is set to True.

```
ru_stage: "{{ora_sw_fs}}/RU19"
```

3.4.17. Installation Precheck true/false. When set to True, during the installation, all the precheck failures will be ignored. Prechecks shouldn’t be ignored in production deployments. Recommended keeping it false.

```
ignoreprecheck: false
```

3.4.18. Fill these parameters when the database must be created on ASM. When set to True, Grid Standalone will be installed and the Diskgroup is created.

```
grid_asm_flag: true
asm_sys_password: oracle
asm_diskgroup_name: DATA
asm_diskgroup_redundancy: EXTERNAL
asm_diskgroup_ausize: 4
asm_diskgroup_disk: ['hdisk2','hdisk3','hdisk4']
asm_disk_loc_prefix: "/dev/r"
asm_diskdiscoverystring: "{{asm_disk_loc_prefix}}hdisk*"
asm_asmsnmp_password: passw0rd
```

3.4.19. Provide the path where the Oracle RDBMS should be installed and the database details. This needs to be updated as a list in the same format as shown below.

```
databases:
  - ora_home: "{{ora_sw_fs}}/19c/database/home"
```

3.4.20. This is the Database section, provide the database name, password & character set.

```
ora_sid: orcl
ora_pwd: passw0rd
ora_character_set: WE8MSWIN1252
```

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'. The header information can be checked 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 the playbook on PowerVS AIX VM we noted that by default, the rootvg (boot) disk size is 20G. Before running the playbook, the rootvg size should be increased by adding a new disk. For example, "extendvg -f rootvg newhdiskX"

3.5. Create/Update ansible.cfg and inventory files inside the collection's playbooks directory. On the managed host (AIX) "/tmp" filesystem is used for ansible remote temporary activities. So, 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
```

3.6. By default, the dnf package will be installed in /opt/freeware/bin. If yum/dnf and python are not configured on the AIX system, execute the bootstrap playbook (demo_bootstrap_dnf.yml) from the ibm.power_aix collection, refer to Appendix section for the steps.

Upon successful configuration of the dnf/yum, the playbook to create the database can be kicked off. Go to the power_aix_oracle <collection directory>/playbooks.

```
$ cd <collection_dir>/playbooks/
$ cat demo_play_aix_oracle.yml
        - hosts: aixhost
          gather_facts: yes
          vars_files: vars/oracle_params.yml
          roles:
            - role: preconfig
              tags: preconfig
            - role: oracle_install_gi
              tags: oracle_install_gi
            - role: oracle_install_db
              tags: oracle_install_db
            - role: oracle_createdb
              tags: oracle_createdb

$ ansible-playbook demo_play_aix_oracle.yml
```

4. The playbook can also be run using the ansible tags for a specific task as shown below:

4.1. To run only the preconfig tasks

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

4.2. To install and configure only the Single Instance GI

```
$ ansible-playbook demo_play_aix_oracle.yml --tags preconfig,oracle_install_gi
```

4.3. To Install only the RDBMS software

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

5. Troubleshooting

In case of any failures during the playbook execution and to debug the problem, try using -vvv option

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

For further support, please log an issue in the Github.

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

6. **Collection Roles:** This collection has seven roles:

6.1. preconfig:

- Expands /var and /opt filesystems
- Running cfgmgr to discover new devices
- Changes /tmp to 8G size to hold ansible temp files
- Changes ulimits for default user to unlimited
- Checks rpm Packages and use dnf for installing rpm packages.
- Setting the DNS
- Checking /etc/hosts file on managed host and adds the entry if needed.
- Changes maxuproc value.
- Sets the OS paging size
- Does the validation of VG & ASM disks.
- Checks and sets the iocp attribute to “available”.
- Creates NFS filesystems for filesets.
- Checks and Installs the Filesets.
- Reboots the lpar if needed

6.2. oracle_install_gi:

- Creates NFS filesystem from nfshost for Oracle Binaries.
- Creates the required GI groups and users.
- Creates the volume group and mounts the file system for the Standalone GI installation
- Creates Oracle installation directories.
- If the RU option is selected, the RU zip and OPatch from NFS/remote/local are extracted.
- Generates the response files and Installs Standalone Grid with the required versions - either base version (19.3) or with RU or with the Gold image along with the option to include or exclude the ignoreprecheck parameter.
- Executes root / config scripts.

6.3. oracle_install_db:

- Creates NFS filesystem from nfshost for Oracle Binaries.
- Creates the required RDBMS groups and users.
- Creates the volume group and mounts the file system for the Standalone RDBMS installation
- Creates Oracle installation directories.
- If the RU option is selected, the RU zip and OPatch from NFS/remote/local are extracted.
- Generates the Oracle response file and Installs RDBMS software with the required versions - either base version (19.3) or with RU or with the Gold image along with the option to include or exclude the ignoreprecheck parameter.
- Executes root scripts.

6.4. oracle_createdb:

- Checks the /etc/oratab file for DB existence
- If grid_asm_flag option is true, then it creates the database on ASM storage
- If grid_asm_flag option is false, then it creates VG and mounts the filesystems for hosting the database files.
- Generates the Database creation template file and the database creation script
- Creates the database
- For JFS DB, it creates and configures the oracle listener
- Checks Oracle PMON background process status
- Adds TNS Entry to tnsnames.ora.
- Sets the database environment variables in the user’s .profile

6.5. powervc_create_aixvm:

- Creates SSH key pair definition
- Creates a VM using PowerVC
- Lists and registers the created VMs
- Creates and attaches the storage volumes and then adds the disks
- Updates the inventory

6.6. oracle_create_gold_image:

- Checks whether the home is registered in the central inventory and fails if it is not.
- Creates gold images for Grid and Oracle homes by checking inventory presence and available space.

6.7. oracle_deinstall:

- Deinstalls the GI / DB homes

Executing Oracle collection using Ansible Automation Platform 2 (AAP2)

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

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

Use the “ansible-builder” utility to build the Container Image. For more info regarding ansible-builder refer the this link

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

This is how the docker file and the requirements.yml file would look like.

```
$ 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 the execution-environment.yml file, use the 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 the 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+rwx /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"]
```

Listing out the image which was created.

```
$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
```

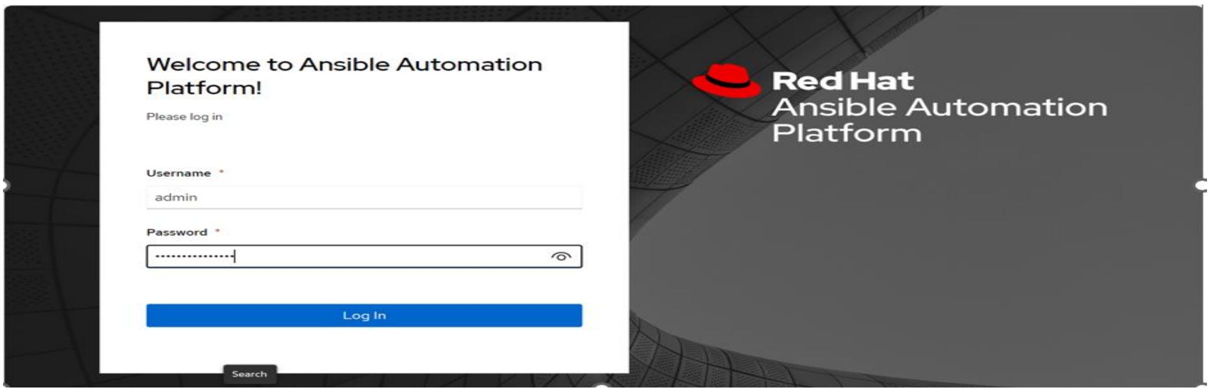
We can use the ansible-navigator utility to execute the playbook via CLI using the execution environments (Container image)

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

PLAY [all] *********************************************************************

TASK [Gathering Facts] *********************************************************
ok: [oracle-aix-test]
```

Executing the Playbook from Ansible Controller AAP2 using Execution Environment via GUI

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



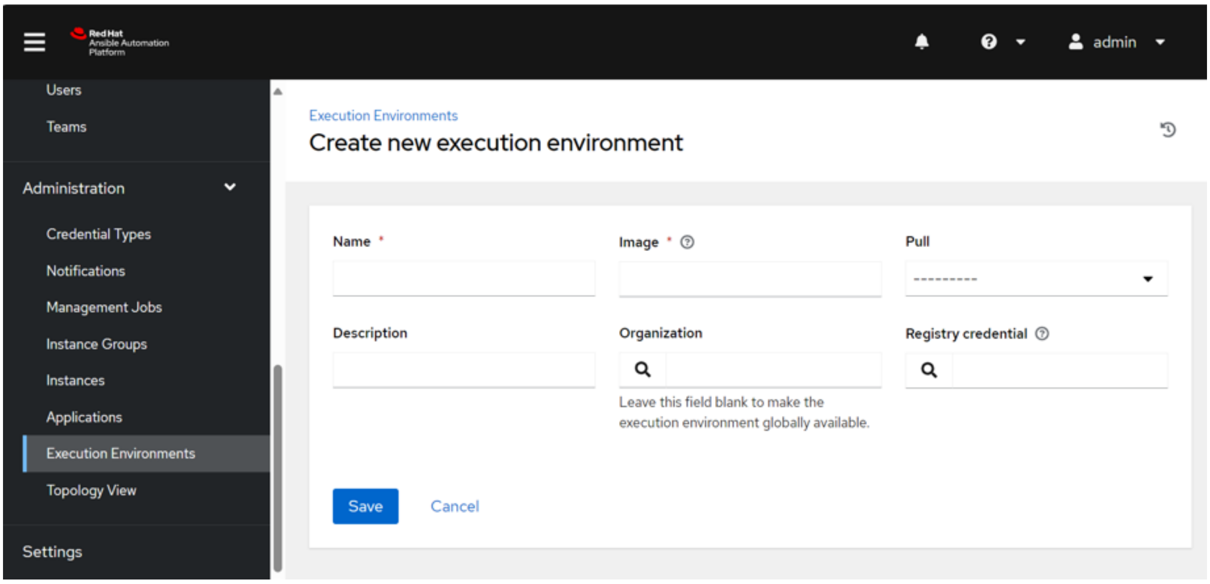
Welcome to Ansible Automation Platform!

Please log in

Username *
admin

Password *
···············

Create a new execution environment: Click on the "Execution Environments" option under the Administration dropdown.



Execution Environments

Create new execution environment

Name *

Image *

Pull

Description

Organization

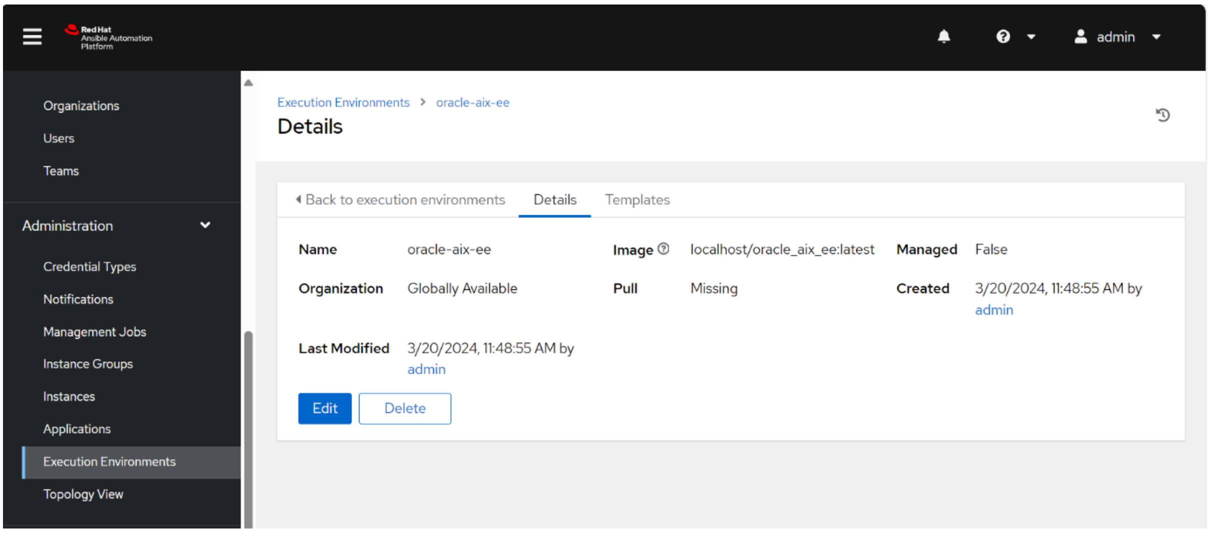
Leave this field blank to make the execution environment globally available.

Registry credential

Fill in 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]

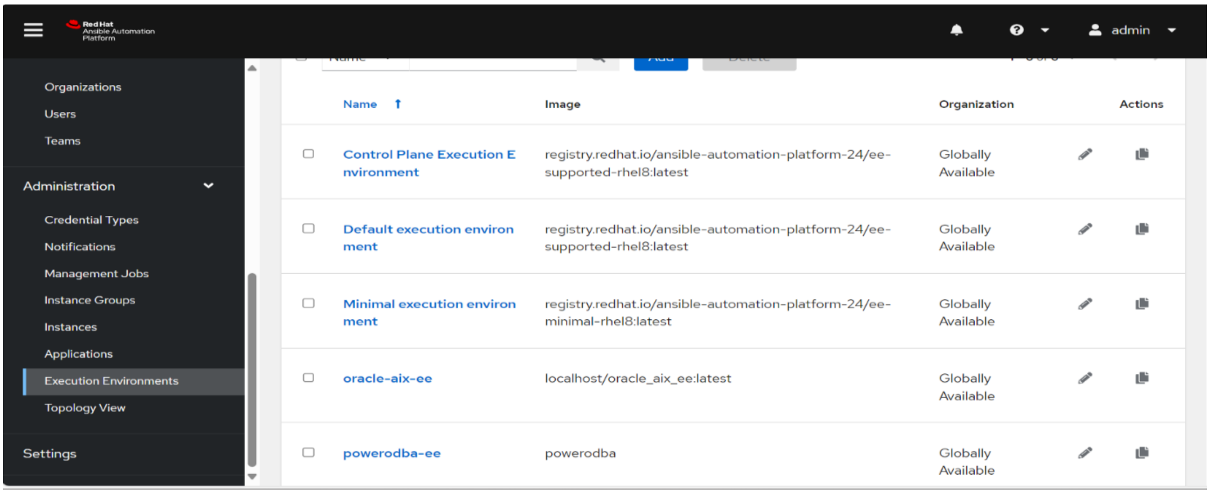
After filling in the details, click on "Save" to create the new execution environment, we can see the following details to crosscheck.



Details

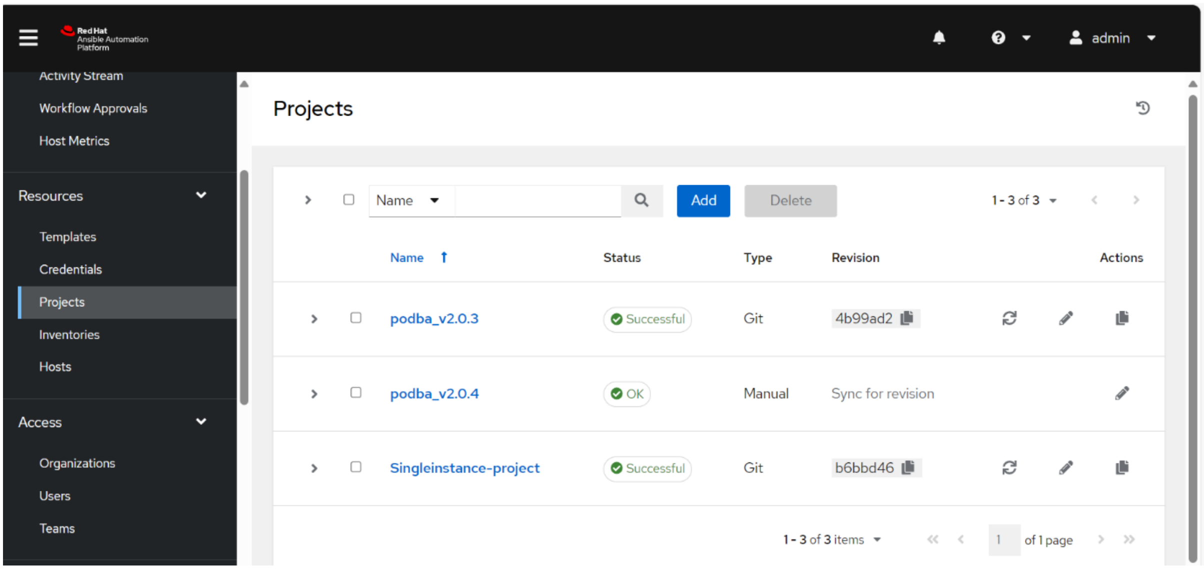
Name	oracle-aix-ee	Image	localhost/oracle_aix_ee:latest	Managed	False
Organization	Globally Available	Pull	Missing	Created	3/20/2024, 11:48:55 AM by admin
Last Modified	3/20/2024, 11:48:55 AM by admin				

Upon saving, the container image will be listed in Execution Environments.



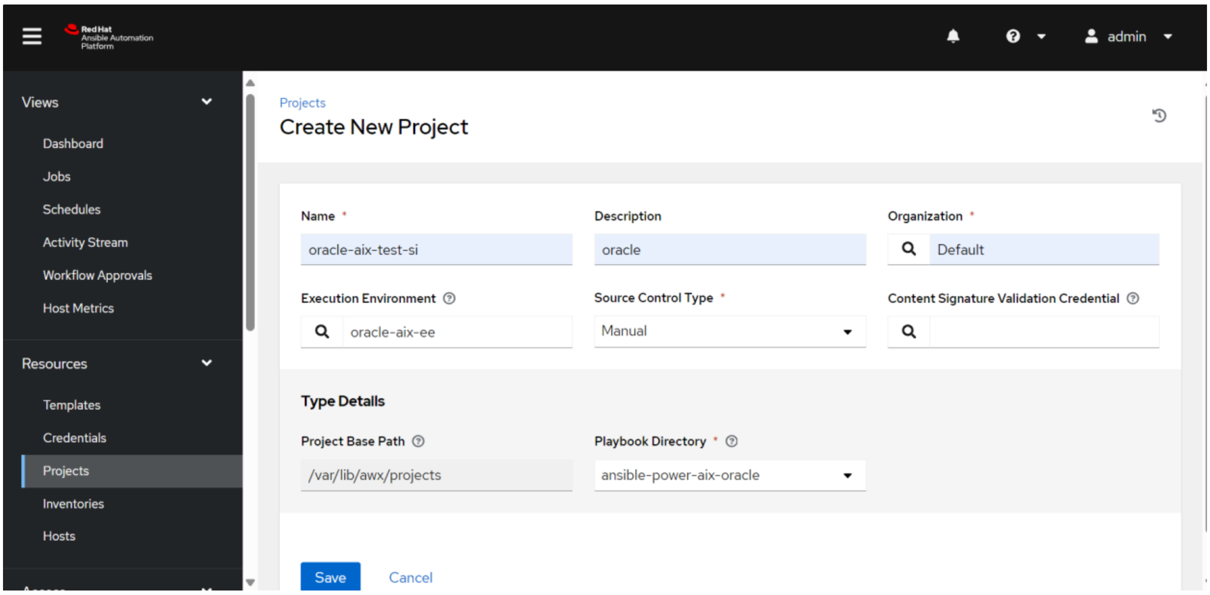
	Name ↑	Image	Organization	Actions
□	Control Plane Execution Environment	registry.redhat.io/ansible-automation-platform-24/ee-supported-rhel8:latest	Globally Available	
□	Default execution environment	registry.redhat.io/ansible-automation-platform-24/ee-supported-rhel8:latest	Globally Available	
□	Minimal execution environment	registry.redhat.io/ansible-automation-platform-24/ee-minimal-rhel8:latest	Globally Available	
□	oracle-aix-ee	localhost/oracle_aix_ee:latest	Globally Available	
□	powerodba-ee	powerodba	Globally Available	

Projects: Click on the "Projects" under the “Resources” dropdown and then click on the "Add" button.



Projects

		Name ↑	Status	Type	Revision	Actions
›	□	podba_v2.0.3	Successful	Git	4b99ad2	
›	□	podba_v2.0.4	OK	Manual	Sync for revision	
›	□	Singleinstance-project	Successful	Git	b6bbd46	



Projects

Create New Project

Name * oracle-aix-test-si

Description oracle

Organization * Default

Execution Environment oracle-aix-ee

Source Control Type * Manual

Content Signature Validation Credential

Type Details

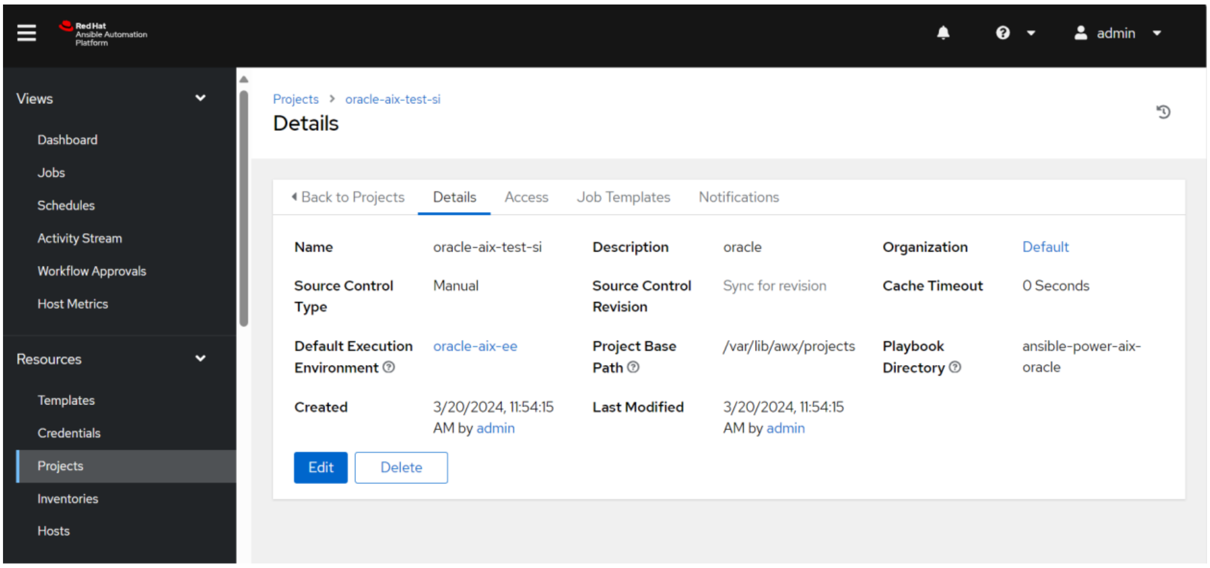
Project Base Path /var/lib/awx/projects

Playbook Directory * ansible-power-aix-oracle

Fill in 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]

Click on "Save"

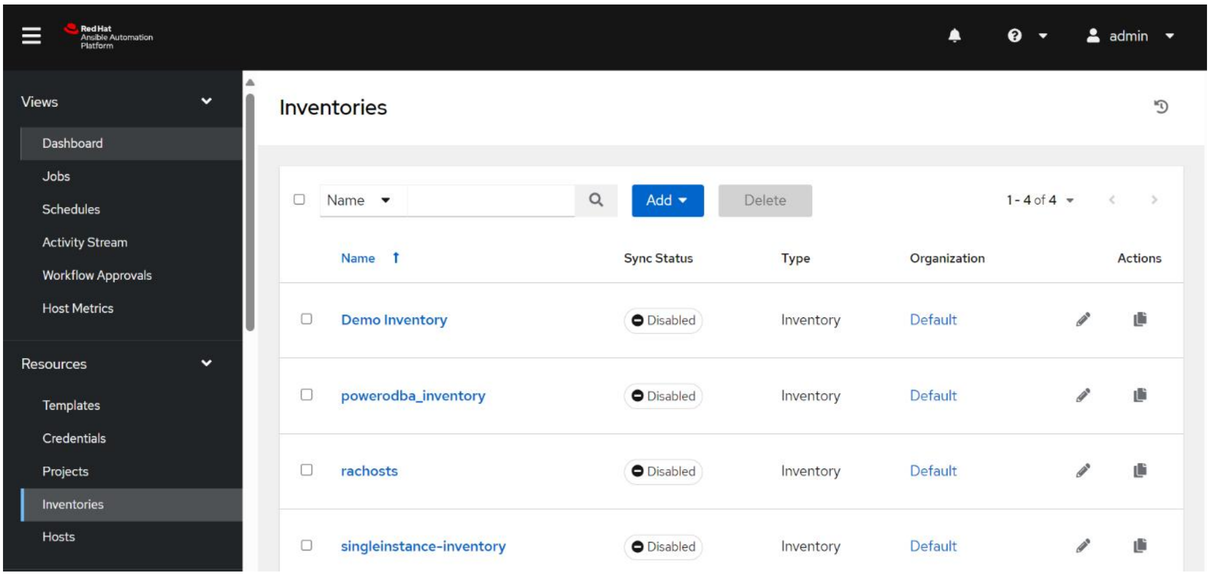


Projects > oracle-aix-test-si

Details

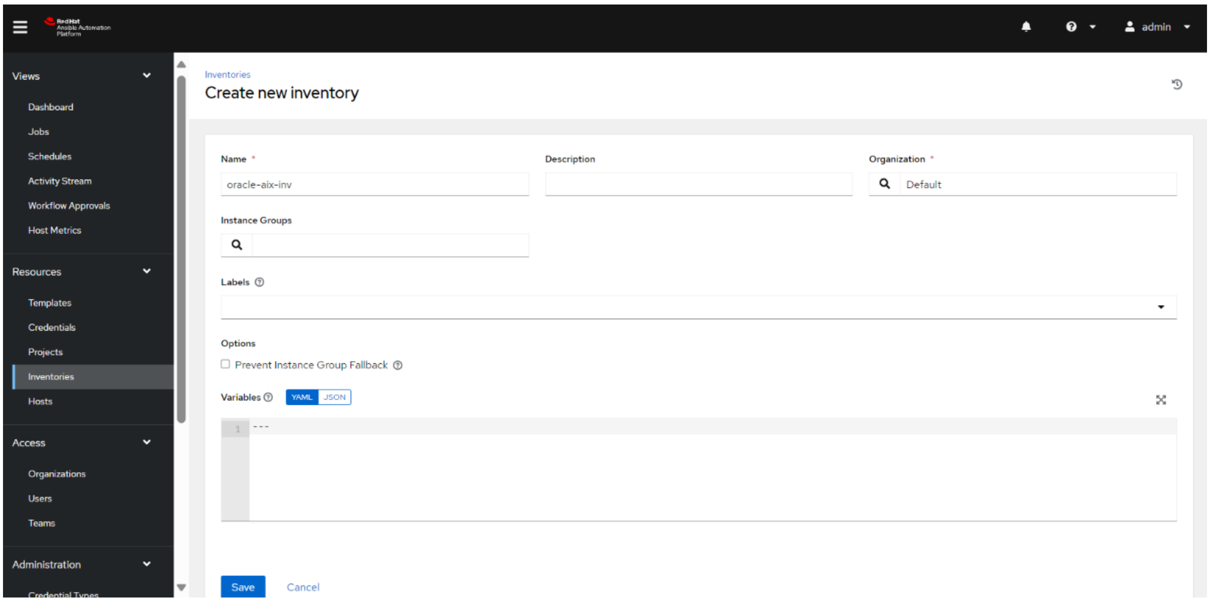
Name	oracle-aix-test-si	Description	oracle	Organization	Default
Source Control Type	Manual	Source Control Revision	Sync for revision	Cache Timeout	0 Seconds
Default Execution Environment	oracle-aix-ee	Project Base Path	/var/lib/awx/projects	Playbook Directory	ansible-power-aix-oracle
Created	3/20/2024,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

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
Demo Inventory	Disabled	Inventory	Default	
powerodba_inventory	Disabled	Inventory	Default	
rachosts	Disabled	Inventory	Default	
singleinstance-inventory	Disabled	Inventory	Default	



Create new inventory

Name * oracle-aix-inv

Description

Organization * Default

Instance Groups

Labels

Options

□ Prevent Instance Group Fallback

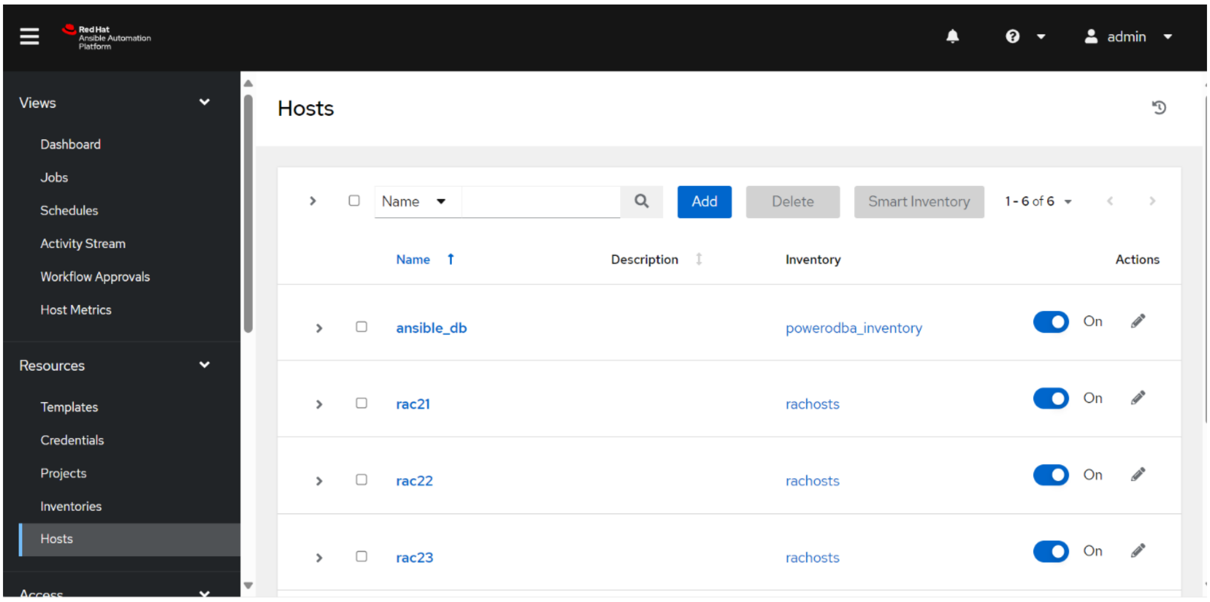
Variables

```
---
```

Fill in the following details:

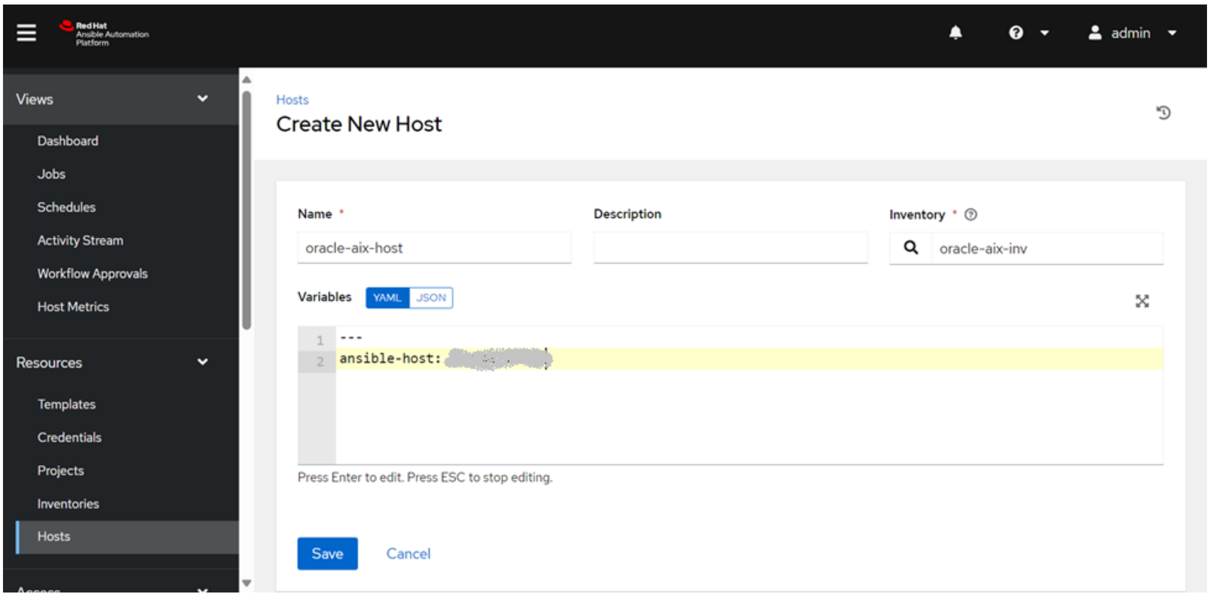
- Name: [Enter the name of your project]
- Description: [Enter a brief description of your project]
- Organization: [Select the organization for your project]

After providing the required information, click on "Save".
Hosts: Click on the "Hosts" option under the “Resources” dropdown and then click on the "Add" button.



Hosts

Name	Description	Inventory	Actions
ansible_db		powerodba_inventory	On
rac21		rachosts	On
rac22		rachosts	On
rac23		rachosts	On



Hosts

Create New Host

Name * oracle-aix-host

Description

Inventory * oracle-aix-inv

Variables

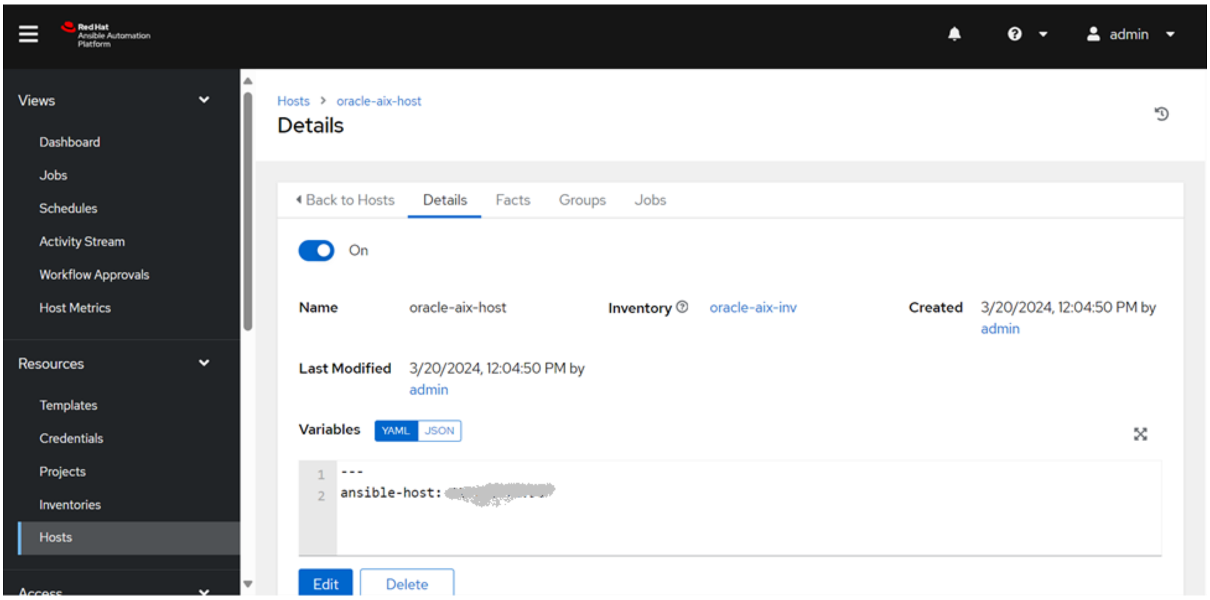
```
---
ansible-host: [redacted]
```

Press Enter to edit. Press ESC to stop editing.

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



Details

On

Name oracle-aix-host

Inventory oracle-aix-inv

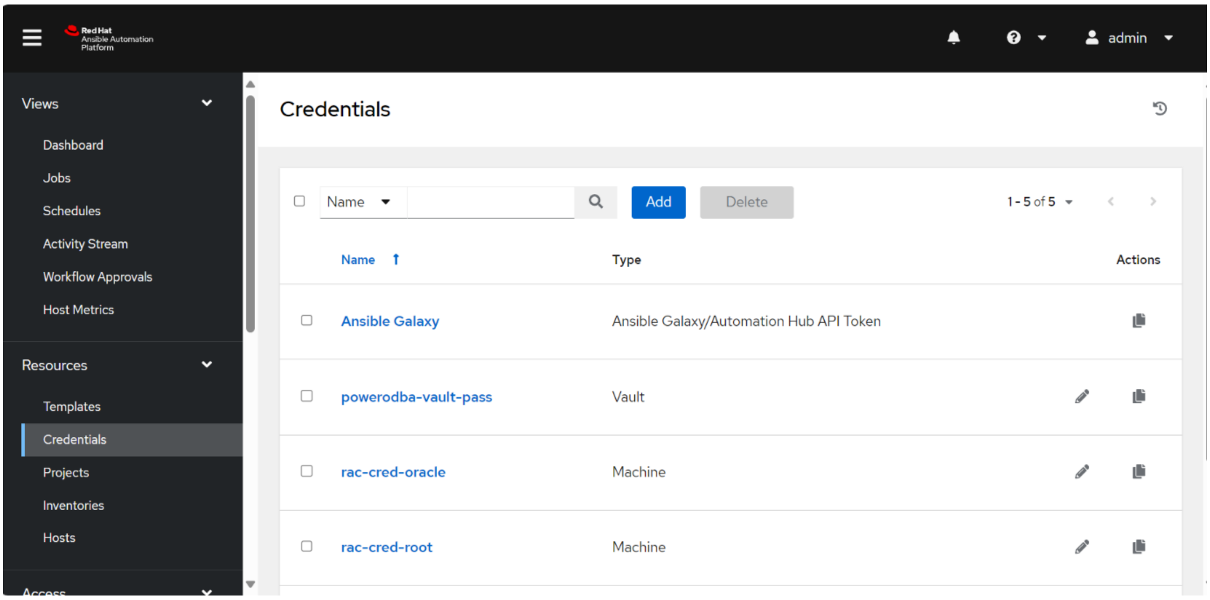
Created 3/20/2024, 12:04:50 PM by admin

Last Modified 3/20/2024, 12:04:50 PM by admin

Variables

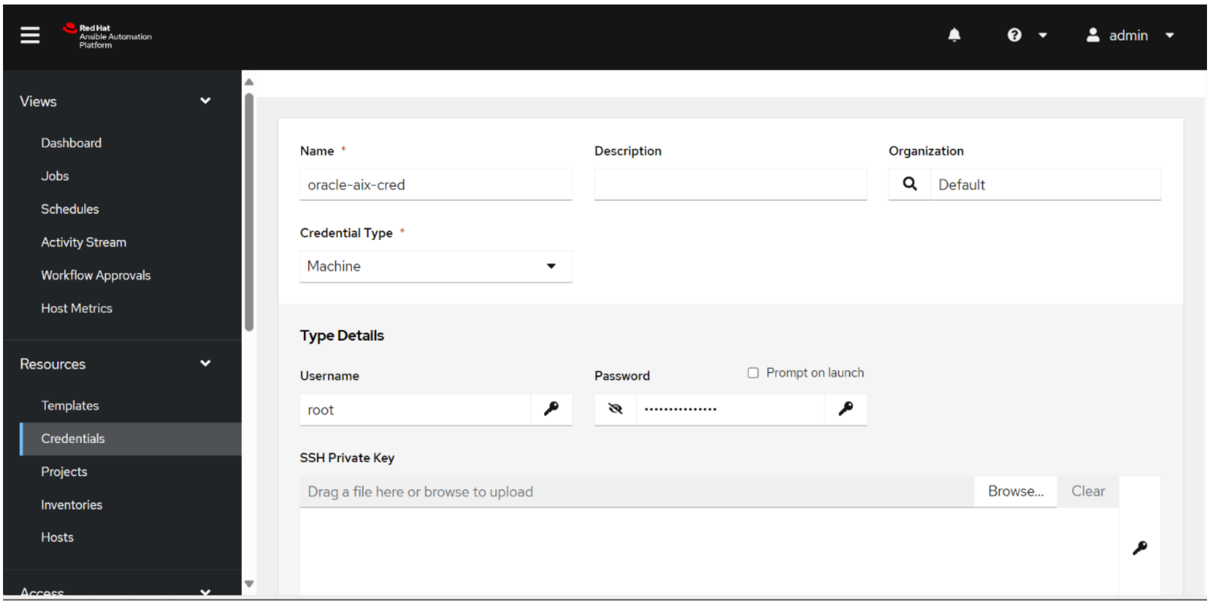
```
---
ansible-host: [redacted]
```

Credentials: Click on the "Credentials" option under the “Resources” dropdown and then click on the "Add" button.



Credentials

	Name	Type	Actions
□	Ansible Galaxy	Ansible Galaxy/Automation Hub API Token	
□	powerodba-vault-pass	Vault	
□	rac-cred-oracle	Machine	
□	rac-cred-root	Machine	



Name * oracle-aix-cred

Description

Organization Default

Credential Type * Machine

Type Details

Username root

Password ••••••••••••••• □ Prompt on launch

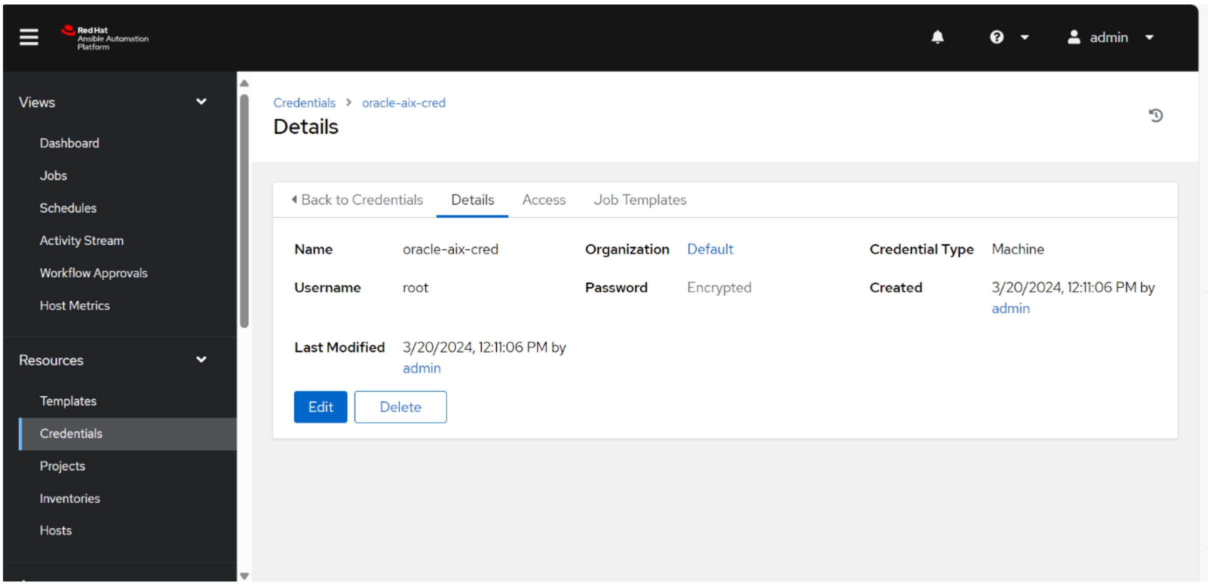
SSH Private Key

Drag a file here or browse to upload

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]

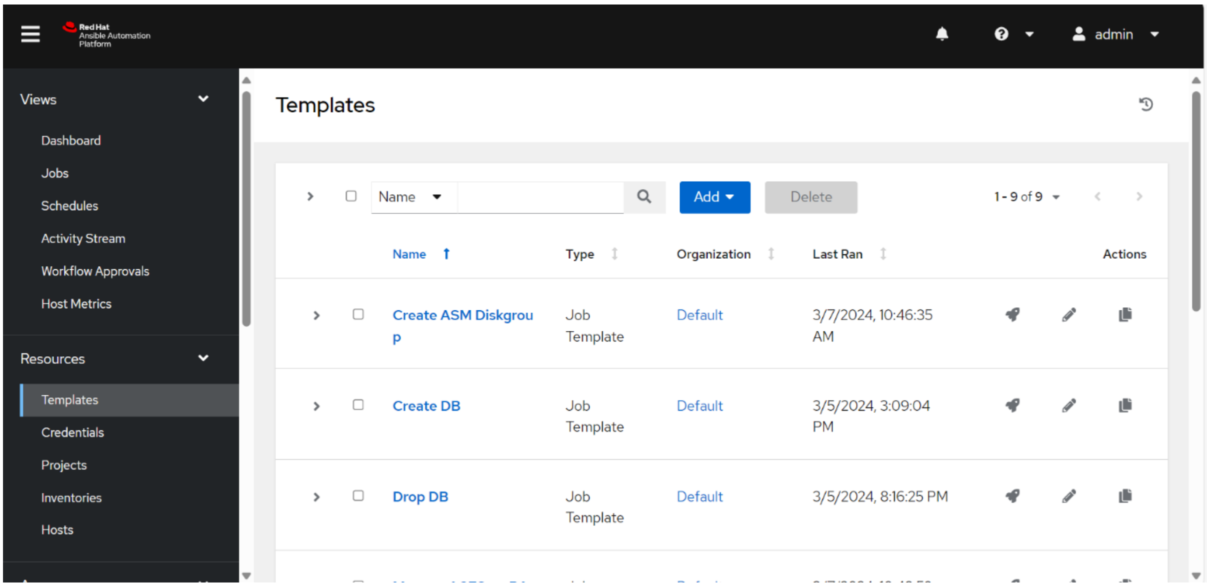
After providing the required information, click on "Save".



Details

Name	oracle-aix-cred	Organization	Default	Credential Type	Machine
Username	root	Password	Encrypted	Created	3/20/2024, 12:11:06 PM by admin
Last Modified	3/20/2024, 12:11:06 PM by admin				

Templates: Click on the "Templates" option under the “Resources” dropdown and then click on the "Add" button.



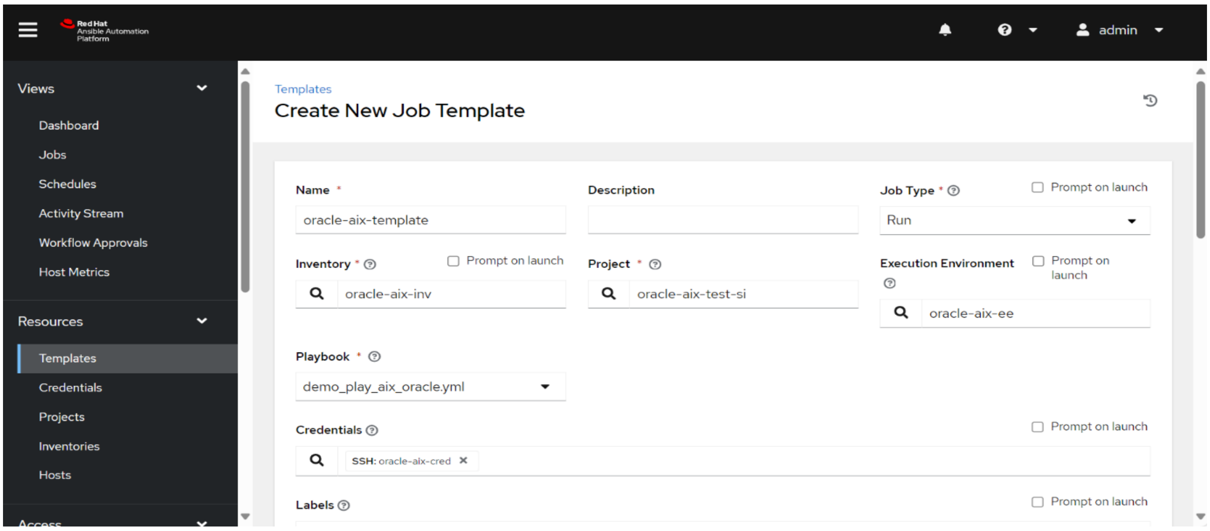
Templates

Name	Type	Organization	Last Ran	Actions
Create ASM Diskgroup	Job Template	Default	3/7/2024, 10:46:35 AM	
Create DB	Job Template	Default	3/5/2024, 3:09:04 PM	
Drop DB	Job Template	Default	3/5/2024, 8:16:25 PM	

Fill in 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 * oracle-aix-inv ☐ Prompt on launch

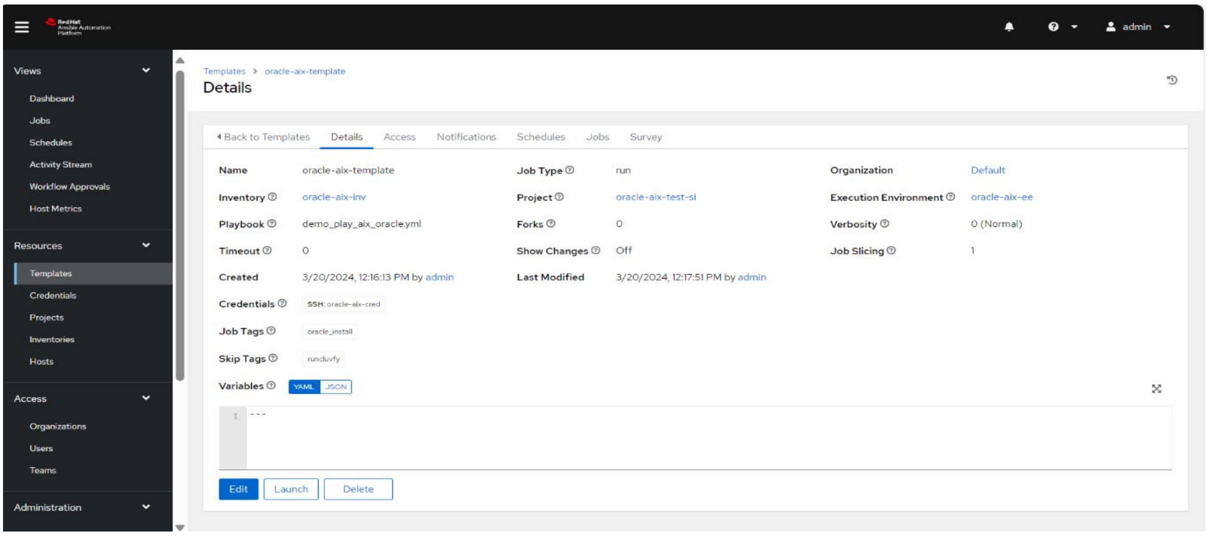
Project * oracle-aix-test-si

Execution Environment oracle-aix-ee ☐ Prompt on launch

Playbook * demo_play_aix_oracle.yml

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

Labels ☐ Prompt on launch

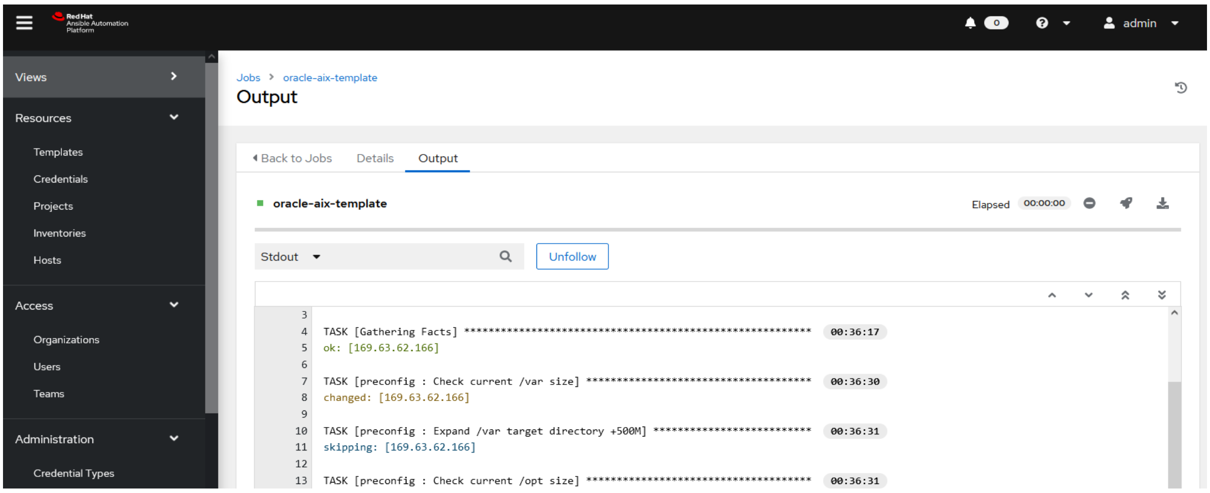


Details

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	runcluvfy				
Variables	YAML				

```
---
```

After providing the required information, click on the "Launch" button. This will launch the job.



Output

oracle-aix-template

```

TASK [Gathering Facts] *********************************************************   00:36:17
ok: [169.63.62.166]

TASK [preconfig : Check current /var size] *************************************   00:36:30
changed: [169.63.62.166]

TASK [preconfig : Expand /var target directory +500M] **************************   00:36:31
skipping: [169.63.62.166]

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.

Prepare 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
```

Authenticate OpenStack Modules with IBM PowerVC

Method 1: These files /opt/ibm/powervc/powervcrc file and /etc/pki/tls/certs/powervc.crt must be copied to the collection directory or any other directory on the 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.

Method 2: 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"
```

Method 3: Define powervc environment in the playbook itself, as shown below 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
…
```

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 with the latest version of OS AIX 72 or AIX 73. Ensure the rootvg disk has 40G of space
2. Install the AIX fileset xlfrte.aix61
3. Install the unzip rpm package
4. If dnf and python3 is not present, run the “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 a new image

Note: If the rootvg disk size is being increased, please make sure to run the "chlv -x 1024 hd6" and "chvg -g rootvg", so that rootvg Total PPs get updated

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 this document.
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
```

Run the Playbook

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

```
$ <collection_dir>/playbooks
$ 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_gi
      tags: oracle_install_gi
    - role: oracle_install_db
      tags: oracle_install_db
    - role: oracle_createdb
      tags: oracle_createdb
```

Run the following commands to execute the playbooks

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

Gold Image Creation

In this section, we'll describe how to create a gold image from an existing setup. Please perfom the steps mentioned in sections 3.1 & 3.2 of this document.

Update variables in variable file of power_aix_oracle collection, <collection dir>/playbook/vars/ oracle_create_goldimage.yml

Below snippet for (both grid and DB) gold image creation.

Set grid_asm_flag to true if a Grid Infrastructure gold image is required (e.g., for ASM-based databases); otherwise, set it to false.

```
work_dir:   /tmp/ansible

ora_user: oracle
ora_oinstall_group: oinstall
ora_inventory: /u01/oraInventory
ora_home: /u01/product/19c/database

# Fill When DB lies on ASM

grid_asm_flag: true
grid_ora_home: /u01/grid/home

# Grid reponsefile parameter
ora_grid_user: grid

# Gold Image destination
goldimage_destination: /u01/gold_image
```

Execute the playbook.

```
$ cat goldimage_play_aix_oracle.yml
---
- name: Playbook
  hosts: all
  gather_facts: yes
  vars_files: vars/gi_oracle_params.yml
  roles:
    - role: oracle_create_gold_image
      tags: oracle_create_gold_image

$ ansible-playbook goldimage_play_aix_oracle.yml
```

Deinstall RDBMS and standalone Grid home:

Update variables in variable file of power_aix_oracle collection, <collection dir>/vars/ deinstall_oracle_params.yml

Below snippet (both grid and DB) gold image creation.

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

```
work_dir:   /tmp/ansible

ora_user: oracle
ora_oinstall_group: oinstall
ora_sw_fs: /u01
ora_inventory: /u01/oraInventory
ora_home: /u01/product/19c/database

# Fill When DB lies on ASM

grid_asm_flag: true
grid_ora_home: /u01/grid/home

# Grid reponsefile parameter
ora_grid_user: grid
```

```
$ cat deinstall_play_aix_oracle.yml

---
- name: Playbook
  hosts: all
  gather_facts: yes
  vars_files: vars/deinstall_oracle_params.yml
  roles:
    - role: oracle_deinstall
      tags: oracle_deinstall

$ ansible-playbook deinstall_play_aix_oracle.yml
```

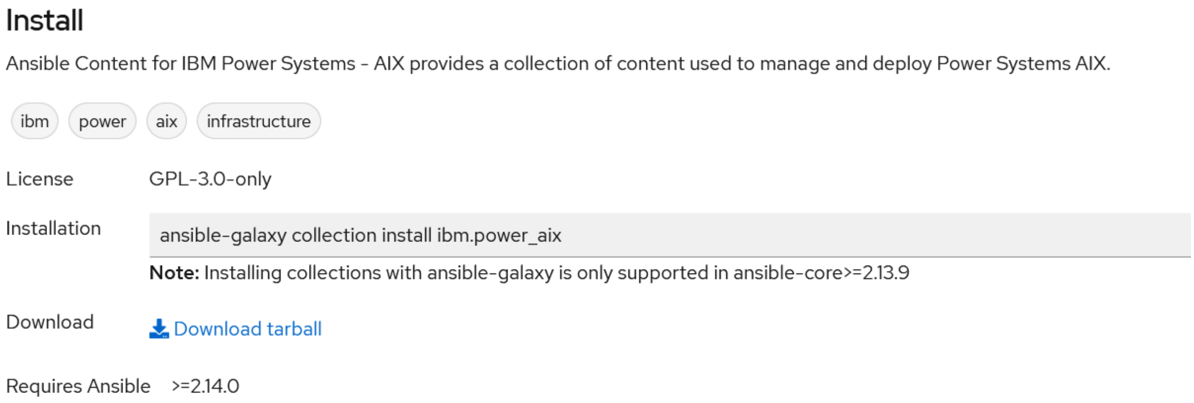
Appendix

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

In AIX7.3/7.2, the python3 and dnf are available at /opt/freeware/bin/ directory. If they are not, the following procedures will help in installing 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.

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

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

1. Prepare the playbook demo_dnf_bootstrap.yml available in "~/.ansible/collections/ansible_collections/ibm/power_aix/playbooks/ demo_dnf_bootstrap.yml"
2. Prepare the ansible.cfg, inventory files for playbook execution. "aixhost" is the AIX managed host mentioned in inventory file. Update the "roles_path" to power-aix collection roles directory.

```
$ 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.cpdir"
```

```
$ 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
```

3. 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