

Ansible Oracle 19c RAC installation for AIX/Power Systems

README - V1.3

June, 2024

Contents

Introduction	3
Assumptions when using collection.....	4
Supported environment.....	5
RAC nodes configuration requirements.....	5
DNS server(s) requirements.....	6
Ansible controller configuration requirements	6
Features of the Collection.....	7
Oracle RAC playbook roles.....	8
Steps for executing the Playbook when Infrastructure Is created manually.....	8
Steps for executing the Playbook with Infrastructure automation using PowerVC.....	10
RAC setup on PowerVS LPARs.....	14
Executing Collection using Ansible Automation Platform 2 (AAP2)	14
Executing the Playbook from Ansible Controller AAP2 using execution environment via GUI.....	17
Appendix	37

Introduction

The Ansible® Oracle® RAC ASM collection installs Oracle RAC 19c on AIX® operating System running IBM® Power® Systems servers. It has been tested on 1, 2, 3, 4, 5, 7 and 8 nodes clusters. The version v1.2.1 collection is tested with powerVC version 2.1.1 for 2-node RAC deployment with AIX 7.3TL0SP1. The collections also tested on 2-node RAC with AIX 7.2TL5 and AIX 7.3TL1SP1(AAP2 GUI)

Setting up an Oracle Application Clusters (RAC) on AIX involves setting up an AIX environment on the hosts that meet the RAC's specific requirements from kernel tunables, network attributes, shared disk attributes, passwordless to user equivalent ssh connections etc. The manual process to accomplish these tasks is tedious and error prone. During the Grid and Database install, the GUI frequently prompts for entering input that ties up the user for a long time. The whole installation can take two days for seasoned users.

With the help of Ansible Oracle RAC ASM collection, it takes typically 5 hours to complete a 4-node RAC installation, a tremendous time saving. It's completely hands-free and can consistently recreate Oracle RACs for other projects. The value of this collection helps your organization to improve significant productivity.

New in Version V1.3.X

- Now the Oracle DB home can be on JFS2 filesystem. Added the “acfs_flag” option. If the flag is set to True, the oracle database home is created on shared ACFS filesystem. If the flag is set to False, the oracle database home is created on AIX JFS2 file system local to each node.
- Added precheck tasks to check the availability of Node VIPs and SCAN VIPs before installing Grid. If these VIPs are in use before the grid install, the task will fail; if they are available, the Grid installation will proceed.
- Added a task to validate SSH equivalence for users grid and oracle before grid install. The task fails if the ssh equivalence is not setup is unsuccessful.
- Added support for executing the Playbook from Ansible Controller AAP2 using execution environment via GUI
- DNF will be used for installing RPM packages; yum has been removed.

New in Version V1.2.X

- Added new roles related powervc for infrastructure automation and supports the 2-node RAC deployment.
- Added “use_ignore_prechecks” flag option to ignore prechecks during oracle software installation
- Added “use_powervs_std_nim” flag option, which eliminates use of nfs if the filesets/NIM location already exist on the LPAR
- Ignore the use of yum/dnf if rpms are already got installed on AIX LPARs.
- Bug fixes

New in Version V1.1.X

- Added support for AIX 7.3
- Added support for Ansible Automation Platform v2 (AAP2)
- Tested on PowerVS LPARs

Assumptions when using collection

- The user is familiar with Ansible and should have at least the basic knowledge on YAML for the purpose of setting up the variables to run the playbook. Refer to Red Hat Ansible Automation Platform Docs at <https://docs.ansible.com/automation.html>
- The user is familiar with AIX administration and Oracle RAC configuration environment requirements. Refer to Oracle Grid Infrastructure Installation and Upgrade Guide 19c for IBM AIX on Power Systems (64-bit) at <https://docs.oracle.com/en/database/oracle/oracle-database/19/books.html>
- The user is familiar with installing packages on the Operating Systems supported by Ansible.

<https://access.redhat.com/articles/3168091>

Below Figure.1 shows the system topology diagram, where the LPARs are manually created from HMC, networks and storage disks are assigned to the LPARs manually. Once the infrastructure is ready, the rac automation playbook is executed from ansible controller.

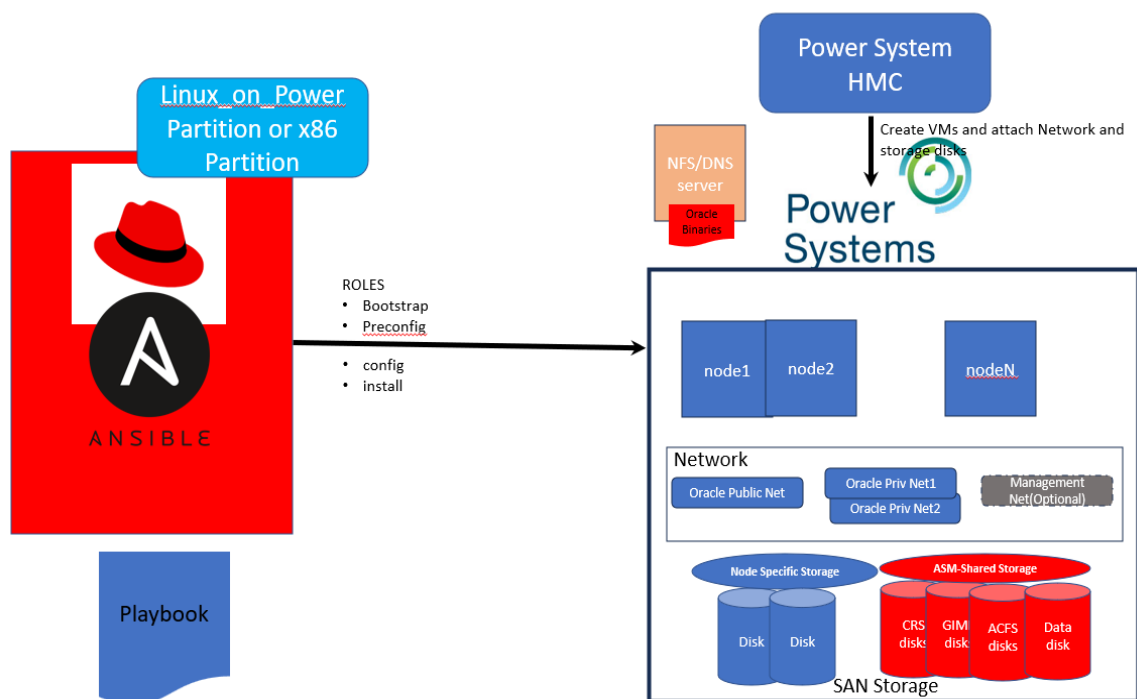


Figure 1. System Topology

Supported environment

- AIX 7.2 TL4 or later, AIX 7.3 TL0 SP1 or later. Tested on AIX 7.2 TL4 SP1, AIX 7.2 TL5 SP2, AIX 7.3 TL0 SP1 and AIX 7.3 TL1 SP1.
- Oracle 19c RAC Standalone Cluster configuration type only. Base release, 19.3 alone is not supported. Base release, 19.3 is supported for install up to 19.17 (using -applyRU option) Tested releases were 19.8, 19.11, 19.12, 19.14 and 19.17.
- Oracle database is installed using Software install on ACFS shared filesystem and JFS2 filesystem.
- Ansible 2.9 or above. Tested versions were Ansible 2.9.11, 2.12.1, 2.15

Note : At the time of writing, Sep 26, 2023, the Base release 19.3 does not work with 19.18 or later and it will fail because of known oracle Bug.

Bug#34962446 19.18 - AIX:OPATCH 35 - DELETEACTION : DESTINATION FILE IS NOT WRITEABLE.

For installing 19.18 or later releases, first the user can use this rac ansible collection scripts to install RU19.17 on top of base release 19.3. Next user can use the PowerODBA collection scripts to upgrade stack to RU19.18 or later.

https://galaxy.ansible.com/ibm/power_aix_oracle_dba

RAC nodes configuration requirements

- Each RAC node should have AIX 7.2 TL4 SP1 or above freshly installed.
- Three types of networks for RAC should be appropriately configured and tested for connectivity. Each RAC node must have:
 - Internet access network for downloading software packages, which connects to AIX toolbox for RPM install. This can be optional if the required rpm packages are already installed before running the playbook.
 - Oracle public networks (if this network can access the Internet, the Internet access network does not need to be configured).
 - Oracle private networks. Two networks are dedicated to Oracle for Interconnect and ASM traffic. Separate ASM networks and Interconnect networks is not supported.

All network interfaces must be consistent across the nodes. E.g. en0 on all nodes are connected to the same physical network, likewise for en1 and en2 etc. The playbook will perform consistency check and connectivity check on Oracle public networks and Oracle Interconnect networks.

- ASM shared disks are free of physical volume IDs (PVIDs), do not belong to any AIX volume groups nor has ASM disk group header. Refer to Appendix **Disk headers and PVIDs** for more information.

- ASM shared disks are consistent across all RAC nodes, for instance, hdisk10 on each node is indeed the same LUN off the storage. The playbook will check for shared disk consistency.
- Oracle Grid HOME is on a local filesystem created out of an AIX volume group, make sure the disks for the volume group do not have previous volume group associated with them.
- AIX filesets(Optional), Opatch zip files, Grid Infrastructure/Database install zip files, IBM XLC compiler(Optional) filesets are available to the RAC nodes through NFS mount points.

DNS server(s) requirements

The RAC environment must have DNS server(s) for Domain Name resolution to resolve Oracle RAC SCAN name, Oracle Virtual IP addresses and for downloading software from the Internet. There are two choices:

- A DNS server that resolves Domain Name addresses for software download from the Internet, Oracle SCAN address, and Oracle Virtual IP addresses.
- Two DNS servers – one for resolving Domain Name addresses for software download from the Internet only, another for Oracle SCAN name and Oracle Virtual IP addresses. The two servers can be on different networks.

Ansible controller configuration requirements

- The Ansible controller has network connectivity to all RAC nodes. It has been tested with the controller and the RAC nodes connected over WAN/VPN as well as within the lab.
- Create a regular user such as 'ansible' for installing Ansible and running the playbook.
- Install python3 and python3-netaddr packages.
- The controller has wget, perl and expect RPMs installed.
- The controller has Ansible 2.9 or above installed. For details visit https://docs.ansible.com/ansible/latest/installation_guide/index.html
- Download the Ansible Oracle RAC collection and create a "top directory" for the collection.
- Install power_aix collection as the user who runs Ansible playbook.
 - \$ ansible-galaxy collection install ibm.power_aix
 - For details, visit <https://ibm.github.io/ansible-power-aix/installation.html>
 - The collection is installed in ~/.ansible/collections/ansible_collections/ibm/power_aix.
- The controller has access to the Internet for downloading software.
- The [openstacksdk](#) must be installed, when using the powervc automation
- On some ansible servers ansible.utils collection is required for the Oracle RAC collection to work. If it is not installed then use *"ansible-galaxy collection install ansible.utils"* to install it.

- Ansible Oracle RAC collection has three files that need to be modified for your environment:
 - `ansible.cfg`
 - `inventory` – specifies the IP addresses of the RAC hosts.
 - When using `powervc` for automating the infrastructure, review the variable files `vars/powervc.yml` & `vars/powervc_rac.yml` and specify the values for variables
 - When you are not using the `powervc` where the `lpars` are created manually, review the variable file `vars/rac.yml` – specifies values for variables to configure the RAC hosts, AIX, Grid Infrastructure and RAC database.

Features of the Collection

- The playbook has been developed with idempotency in mind. Idempotent means a task operation is only performed once, regardless the task is invoked many times thereafter. However, if a configuration variable that affects the outcome of the operation has changed after the operation has already been performed, it will perform the operation in most cases but not all. For instance, the playbook allows disks to be renamed to better indicate their intended purposes, but if the tasks prior to the renaming disks task need to be rerun maybe due to the disks have been renamed incorrectly. In this scenario, tasks that reference renamed disks will fail, and manual intervention is needed. Accommodating changes like this is a best effort attempt, therefore it's crucial to have the correct values for the variables the first time.
- As mentioned above, AIX `hdiskX` can optionally be renamed to `<prefix>X` or `<prefix>Y`, where `Y` is a different number from `X` and `<prefix>` is a user-specified string. This feature makes it easier to correlate the purpose of the disks in a large disk configuration.
- The recommended AIX tunable values and various settings are applied according to applicable Oracle documentation for running a RAC in the areas of networking, kernel, and disk attributes etc. They are built into the Ansible tasks. Refer to Oracle Grid Infrastructure Installation and Upgrade Guide 19c for IBM AIX on Power System (64-bit) at <https://docs.oracle.com/en/database/oracle/oracle-database/19/books.html>
- The `dsm` fileset for distributed shell (`dsh`) will be installed to ease the cluster administrative tasks. The collection also depends on it.
- IBM XL C 13 for AIX compiler installation is optional. It is installed on the named RAC host(s) specified in `vars.yml`. It requires the filesets to be accessible through NFS mount point.
- VNC server RPM will be installed on the RAC nodes when it is specified(optional). Users other than 'root' in the `vars.yml` will have VNC files created in `~/vnc/passwd`.
- Ansible uses `ssh` passwordless connections to drive the setup tasks on the RAC nodes. The bootstrap role (see below) creates the `ssh` passwordless configuration without the need to manually perform it beforehand. Ansible relies on the host's IP addresses and the root login password specified on the `var.yml` to create such configuration.

- AIX system files /etc/environment, /etc/hosts, /etc/security/limits, /etc/pam.conf and /etc/syslog.conf etc. are saved before they are modified. They are saved in <Ansible work directory>/saved directory, where <Ansible work directory> is specified in global section of vars.yml.
- When nim mount is already exists in LPAR then there is no need of NFS server for installing filesets. The LPARs created with powervs standard images have /usr/sys/inst.images filesystem mounted. If you want to install filesets from a specific location set "use_powervs_std_nim" to true and specify the filesets location "powervs_loc" in vars/rac.yml file

```
use_powervs_std_nim: true
powervs_loc: '/usr/sys/inst.images/installp/ppc'
```

- If you want to ignore known prechecks failures during oracle installation, you can the set flag "use_ignore_prechecks" to true in vars/rac.yml file. The oracle installer uses "-ignorePrereqFailure" option during installation

```
use_ignore_prechecks: true
```

Oracle RAC playbook roles

The RAC automation playbook is divided into the four roles:

- bootstrap – sets up the basic environment to enable full functionality of Ansible, set nameserver, binding and passwordless connections to the RAC nodes.
- preconfig – sets up basic environment such as time of day, configure for accessing Internet and consistent AIX version, release, TL, and SP. NFS mounts AIX filesets and installs the filesets.
- config – sets up AIX to meet the requirements for installing a RAC.
- Install – creates ASM disk groups, ACFS, prepares for installing Grid and database and finally install GRID_HOME on JFS2 filesystem and Database ORACLE_HOME on ACFS shared Filesystem/JFS

Steps for executing the Playbook when Infrastructure Is created manually

Architecture is same as shown in Figure 1.

- 1) Download the collection from ansible galaxy. Go to collection directory. Review and update the ansible.cfg file and inventory file
https://galaxy.ansible.com/ibm/power_aix_oracle_rac_asm

- 2) Ensure that the system meets the requirements of ansible controller and rac nodes
- 3) Gather the details necessary for filling the vars/rac.yml variable file

Network	Minimum 3 networks are needed (net1: ora_rac_public, net2: ora_priv1, net3: ora_priv2) Each node will have one node-vip runs on net1 3 Scan-VIPS for cluster runs on net2 All the network interface names should be same across all nodes in the cluster
SAN Storage volumes	Node specific storage volumes are need for staging the oracle Binaries Shared Storage disks are need for ASM Diskgroups(OCR_VOTE, GIMR, ACFSDG and DATA)
DNS server	The nameserver details are needed which will be updated in /etc/resolv.conf file of cluster nodes during playbook execution.
NFS Server	NFS server details are need which will provide the AIX Filesets and oracle software binaries
NTP server(optional)	NTP server will help is synchronizing the time on cluster nodes. If the time on nodes are in sync, ntp server details are optional

- 4) Sample values were given to variables in vars/rac.yml file. Review the variable file and update the variables based on your environment.
- 5) Update the install_and_configure_Oracle_RAC.yml playbook shown below
 - Uncomment the hosts: line and set the field by specifying the inventory group name
 - Uncomment the first variables file (named vars/rac.yml) to have its variables included in this execution.

```
$ cat install_and_configure_Oracle_RAC.yml
---
# install_and_configure_Oracle_RAC.yml
# Powervc based deployments uses variable files vars/powervc.yml,vars/powervc_rac.yml
# If the LPARs are build manually, to automate oracle RAC deployment use variable file vars/rac.yml
#- hosts: "{{ racName }}" # racName variable is defined when you use the powervc automation scripts for building the AIX LPARs
- hosts: orac           # Get the group name from inventory file which contains the oracle cluster nodes
  gather_facts: no
  vars_files:
    - vars/rac.yml
  # - vars/powervc.yml
  # - vars/powervc_rac.yml
  roles:
    - role: bootstrap
  vars:
    download_dir: "~"
    target_dir: "/tmp/.ansible.cpidr"
  tags: bootstrap
  - role: preconfig
```

```
tags: preconfig
- role: config
tags: config
- role: install
tags: install
```

6) Execute the Playbook

ansible-playbook install_and_configure_Oracle_RAC.yml 2>&1 | tee play.out

You can also run single or multiple roles using tags option

```
ansible-playbook install_and_configure_Oracle_RAC.yml --tags bootstrap 2>&1 | tee play_bootstrap.out
```

```
ansible-playbook install_and_configure_Oracle_RAC.yml --tags preconfig 2>&1 | tee play_preconfig.out
```

```
ansible-playbook install_and_configure_Oracle_RAC.yml --tags config,install 2>&1 | tee play.out
```

Some the roles runs for longer duration, you can run the playbook in background using

nohup

```
nohup ansible-playbook install_and_configure_Oracle_RAC.yml > play.out 2>&1 &
```

When rerunning the failing role, it is helpful to use **-vvv** debug option to obtain more debug information with more readable STDOUT and STDERR messages.

Suppose Ansible failed in config role, and the issue is fixed, rerun starting from config role using the **-tags** option like this:

```
$ ansible-playbook -vvv install_and_configure_Oracle_RAC.yml --tags config,install 2>&1 | tee play.out
```

Ansible will run the config and install roles.

Note: While installing openssl dependent packages/rpms using yum or dnf it will fail if openssl version is < 1.1. Refer to Appendix for more details.

The bootstrap role uses the code from power_aix collection, if any issues seen while configuring the dnf/yum/python refer to known issues of power_aix collection

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

It is recommended to use the latest version on power_aix collection

Steps for executing the Playbook with Infrastructure automation using PowerVC

Using the ansible openstack modules the infrastructure creation can be automated on PowerVC. The PowerVC version that is currently supported is 2.1.1. The PowerVC provides the capability of creating the new OS image which from existing. The new image can have the AIX filesets and RPM packages that are required for Oracle RAC. Currently with powervc automation we support only the 2-node oracle RAC setup.

The power_aix_oracle_rac_asm collection contains the following 5 roles pertaining to the infrastructure layer provisioning:

- **powervc_create_network_ports:** creates an openstack port to be used during the node creation. A port defines the IP address and network to be used for a given interface.

- **powervc_create_nodes_without_rac_volumes:** It uses parameters set in the vars/powervc.yml file to create the new cluster nodes.
- **powervc_obtain_token:** obtains a PowerVC access token to establish a REST API connection from Ansible server to PowerVC server. The subsequent ASM disks creation role requires REST API access, hence the need for the token.
- **powervc_create_and_multiattach_asm_volumes:** creates the ASM disks one at a time. Upon creating each disk, this role attaches it to all nodes and runs cfgmgr to ensure the disk maintains the same hdisk number in all nodes as required by Oracle RAC installer.
- **powervc_add_nodes_to_inventory:** update the inventory file with the nodes and additional parameters to set it up for ansible management. This role prepares the environment for execution of the second playbook that is responsible for grid and database software installation.

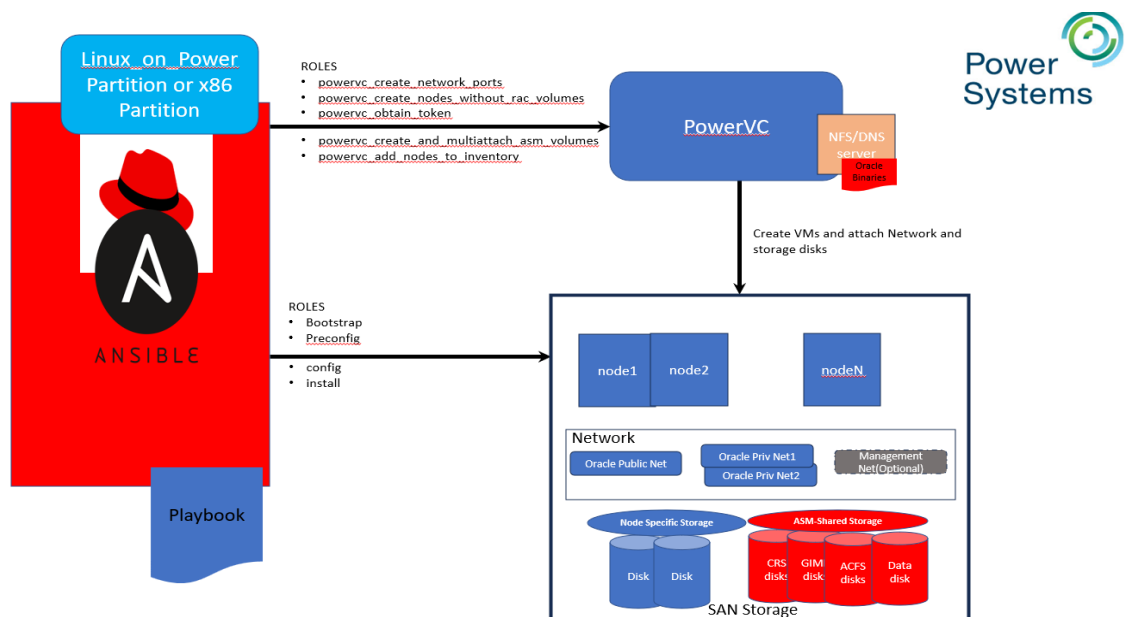


Figure 2. System Topology – Infrastructure automation using PowerVC

Below are the steps that needed to be followed while doing Infrastructure automation for 2-node oracle RAC deployment

- 1) Download the collection from ansible galaxy. Go to collection directory. Review and update the ansible.cfg file
https://galaxy.ansible.com/ibm/power_aix_oracle_rac_asm
- 2) Ensure that the system meets the requirements of ansible controller and rac nodes
- 3) Gather the details necessary for filling the vars/powervc.yml variable file

Network	<p>Minimum 3 networks are needed (net1: ora_rac_public, net2: ora_priv1, net3: ora_priv2) Each node will have one node-vip runs on net1 3 Scan-VIPS for cluster runs on net2</p>
---------	--

	All the network interface names should be same across all nodes in the cluster
SAN Storage volumes	Node specific storage volumes are need for staging the oracle Binaries Shared Storage disks are need for ASM Diskgroups(OCR_VOTE, GIMR, ACFSDG and DATA)
DNS server	The nameserver details are needed which will be updated in /etc/resolv.conf file of cluster nodes during playbook execution.
NFS Server	NFS server details are need which will provide the AIX Filesets and oracle software binaries
NTP server(optional)	NTP server will help is synchronizing the time on cluster nodes. If the time on nodes are in sync, ntp server details are optional

- 4) Sample values were given to variables in vars/powervc.yml file. Review the variable file and update the image section in the vars/powervc.yml file in the collection with the image, image_aix_version and image_password with the latter set to the AIX root password value. Update the other variables that are need for the execution. Also review the vars/powervc_rac.yml variable file as the most of the values are defined from vars/powervc.yml file.
- 5) Optionally, using powervc you can create a new image which contain all required AIX filesets, rpm packages and other required settings. Since some of the ansible tasks are already defined in the new image, during the execution these tasks will get skipped and playbook execution time will get reduced.
- 6) Copy PowerVC certficate file from /etc/pki/tls/certs/powervc.crt file in the PowerVC server to the ansible server to be used in the next step.
- 7) Copy the /opt/ibm/powervc/powervcrc file from the PowerVC server to the ansible server, update its OS_CACERT to where you copied PowerVC certificate file to. Also update it with the user id and password of the PowerVC server and source it.
- 8) Update the install_and_configure_Oracle_RAC.yml playbook shown below
 - Uncomment the hosts line and racName is supplied during playbook execution.
 - Uncomment the variables files (named vars/powervc.yml and vars/powervc_rac.yml) to have its variables included in this execution.

```
$ cat install_and_configure_Oracle_RAC.yml
---
# install_and_configure_Oracle_RAC.yml
# Powervc based deployments uses variable files vars/powervc.yml,vars/powervc_rac.yml
# If the LPARs are build manually, to automate oracle RAC deployment use variable file vars/rac.yml
- hosts: "{{ racName }}"
#- hosts: orac
gather_facts: no
vars_files:
```

```
# - vars/rac.yml
- vars/powervc.yml
- vars/powervc_rac.yml
roles:
- role: bootstrap
vars:
  download_dir: "~"
  target_dir: "/tmp/.ansible.cpubdir"
tags: bootstrap
- role: preconfig
tags: preconfig
- role: config
tags: config
- role: install
tags: install
```

- 9) Review the vars/powervc.yml file and execute the playbook *"powervc_build_AIX_RAC_nodes.yml"* for automatic creation of 2 AIX Lpars and then automate the Oracle RAC installation.

```
---
- name: Build and configure the RAC nodes using PowerVC
  # Must update the vars/powervc.vars var file then call this as follows:
  # ansible-playbook build_AIX_nodes_for_Oracle_RAC.yml -e racName=<theRACname>"
  # e.g. ansible-playbook build_AIX_nodes_for_Oracle_RAC.yml -e racName=orac
  hosts: localhost

  tasks:
  - include_vars: "vars/powervc.yml"
  - fail:
    msg: "racName is required for this playbook to build a dual-node Oracle RAC."
    when: racName is not defined

  - name: Display the input name prefix and count of VMs to be built
    debug:
      msg: "Creating nodes {{racName}}1 and {{racName}}2 for this dual-node Oracle RAC."

  - name: define the network ports based on the networks and IP addresses to be used.
    import_role: name=powervc_create_network_ports

  - name: Create new AIX VMs to act as Oracle RAC nodes
    import_role: name=powervc_create_nodes_without_rac_volumes

  - import_role: name=powervc_obtain_token
  - include_role: name=powervc_create_and_multiattach_asm_volumes
    with_items: "{{ disks }}"

  - name: Now the nodes are good to go, add them to the inventory file to be managed by Ansible
    import_role: name=powervc_add_nodes_to_inventory

  # Importing the playbook to be used for installing and configuring the Oracle RAC.
  - import_playbook: install_and_configure_Oracle_RAC.yml
```

ansible-playbook powervc_build_AIX_RAC_nodes.yml -e racName=orac

Here the racName is given as orac, the playbook will create 2 nodes named as orac1 and orac2.

If you want to execute the playbook "install_and_configure_Oracle_RAC.yml" separately you can comment the import_playbook line. Later you can execute the rac playbook using below command

```
ansible-playbook install_and_configure_Oracle_RAC.yml -e racName=orac
```

RAC setup on PowerVS LPARs

- 1) Create LPARs that are need for RAC cluster with Networks that are needed for oracle RAC
- 2) Create small size LPAR, configure DNS and NFS – to stage oracle software
- 3) Set the root user password for RAC nodes
- 4) Add Local and shared storage LUNs that are needed for oracle RAC installation
- 5) Extend the rootvg volume group, In PowerVS LPAR the boot LUN size is fixed AIX72/71 – 20G and AIX73 - 25G. This is need to accommodate swap space and ansible remote location requirements. Here hdiskX is the newly added disk, this disk size can be > 50G.

```
chvg -t 4 rootvg  
extendvg -f rootvg hdiskX  
chlv -x 1024 hd6
```

- 6) Fill the vars/rac.yml file and Execute the play book

Executing 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 how to create the containerized image and execute the playbook using execution environment(Containerized image).

At first "ansible-builder" is used to build the Container Image

For more info regarding ansible-builder refer to below

- https://docs.redhat.com/en/documentation/red_hat_ansible_automation_platform/2.1/html/ansible_builder_guide/indexCreate the execution-environment.yml

```
$ cat execution-environment.yml

# Execution Environment
---
version: 3
images:
  base_image:
    name: registry.redhat.io/ansible-automation-platform-24/ee-minimal-rhel8:latest
options:
  package_manager_path: /usr/bin/microdnf
additional_build_steps:
  append_base:
    - RUN microdnf install dnf
    - RUN microdnf install python3-pip
    - RUN /usr/bin/pip3 install netaddr
    - RUN microdnf install expect
    - RUN microdnf install perl
dependencies:
  galaxy: requirements.yml

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

- Create container image using ansible builder

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

$ podman images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
localhost/oracle_rac_aix_ee	latest	8d597e0a1bab	About a minute ago	493 MB
localhost/oracle_aix_ee	latest	a7d4b0992ea4	2 months ago	475 MB
localhost/powerodba	latest	3d1b75b3ee16	3 months ago	910 MB
registry.redhat.io/ansible-automation-platform-24/ee-supported-rhel8	latest	b2d26de2d8de	11 months ago	1.79 GB
registry.redhat.io/ansible-automation-platform-24/ee-minimal-rhel8	latest	c239714e9480	11 months ago	380 MB
quay.io/ansible/ansible-runner	latest	bec0dc171168	2 years ago	816 MB

- A context directory is created, and we can see the Container file inside it. Displaying the content of Container/Docker file

\$ 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
RUN microdnf install python3-pip
RUN /usr/bin/pip3 install netaddr
RUN microdnf install expect
RUN microdnf install perl
# 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"]
```

- ansible-navigator is used for executing the playbook in CLI using execution environments (Container image). Go to power_aix_oracle_rac_asm collection and create ansible-navigator.yml file

```
$ cat ansible-navigator.yml
---
ansible-navigator:
  execution-environment:
    container-engine: podman
    enabled: True
    environment-variables:
      set:
        ANSIBLE_CONFIG: ansible.cfg
    image: localhost/oracle_rac_aix_ee:latest

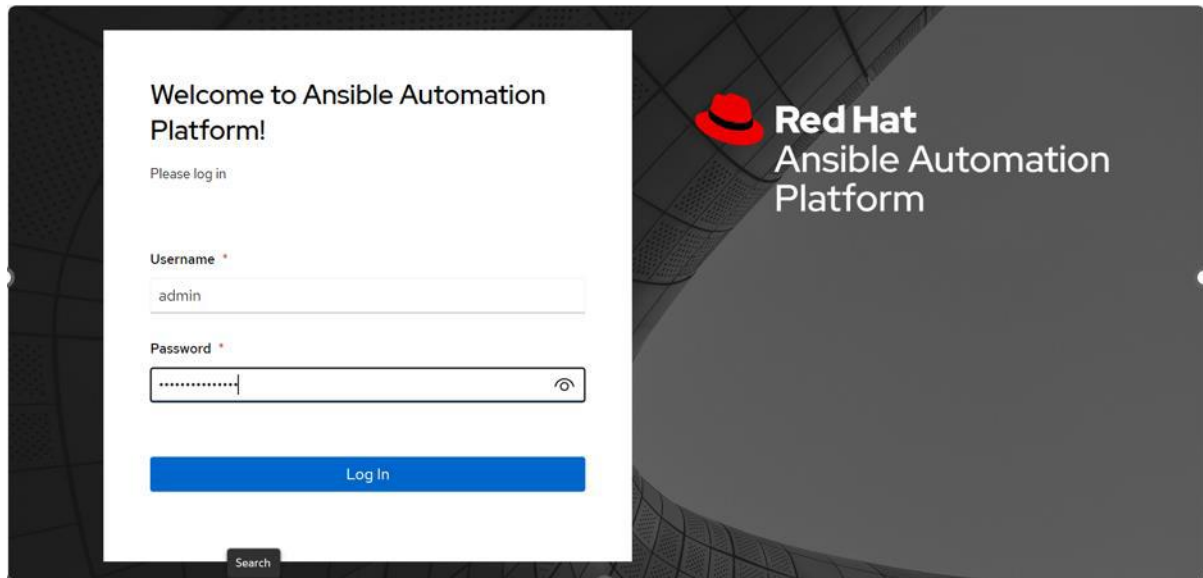
$ ansible-navigator run play.yml --pp=missing -m stdout 2>&1 | tee play_aap2.out
```

For more details regarding ansible-navigator refer to below link

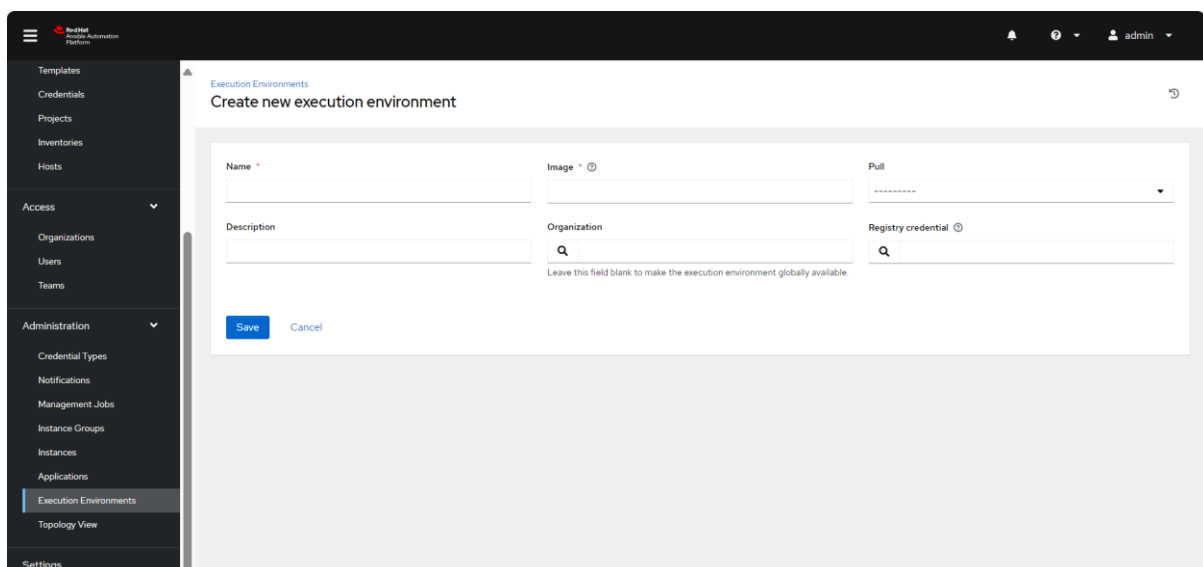
<https://ansible.readthedocs.io/projects/navigator/>

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 and click on Add



Please provide the following details to create a new execution environment:

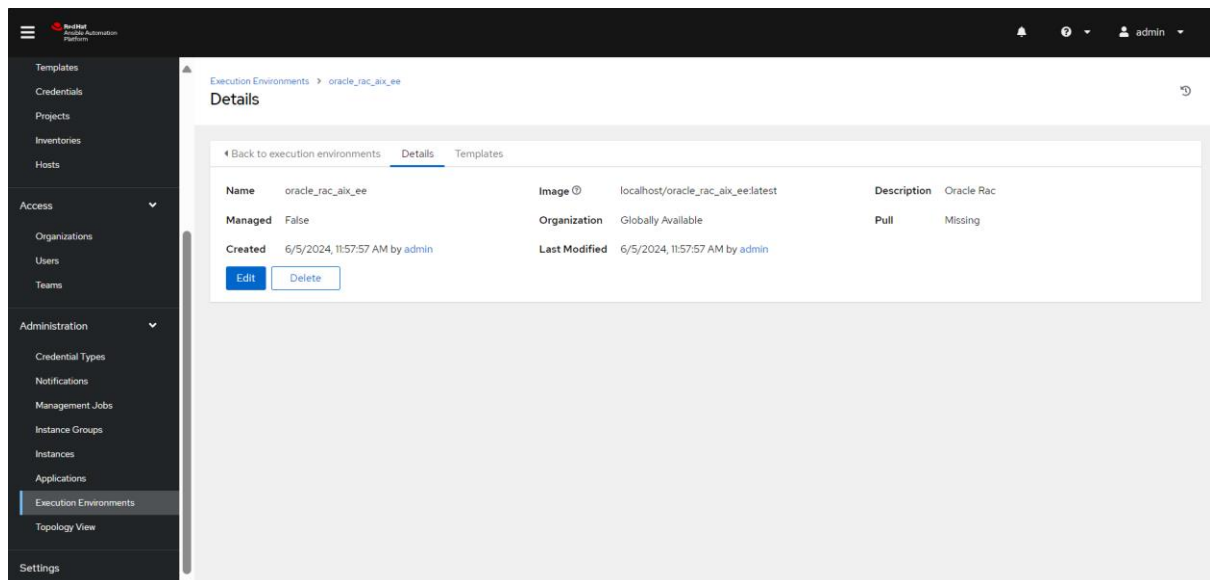
Name: [Enter the name of the execution environment]

Image: [Specify the Docker image for the execution environment]

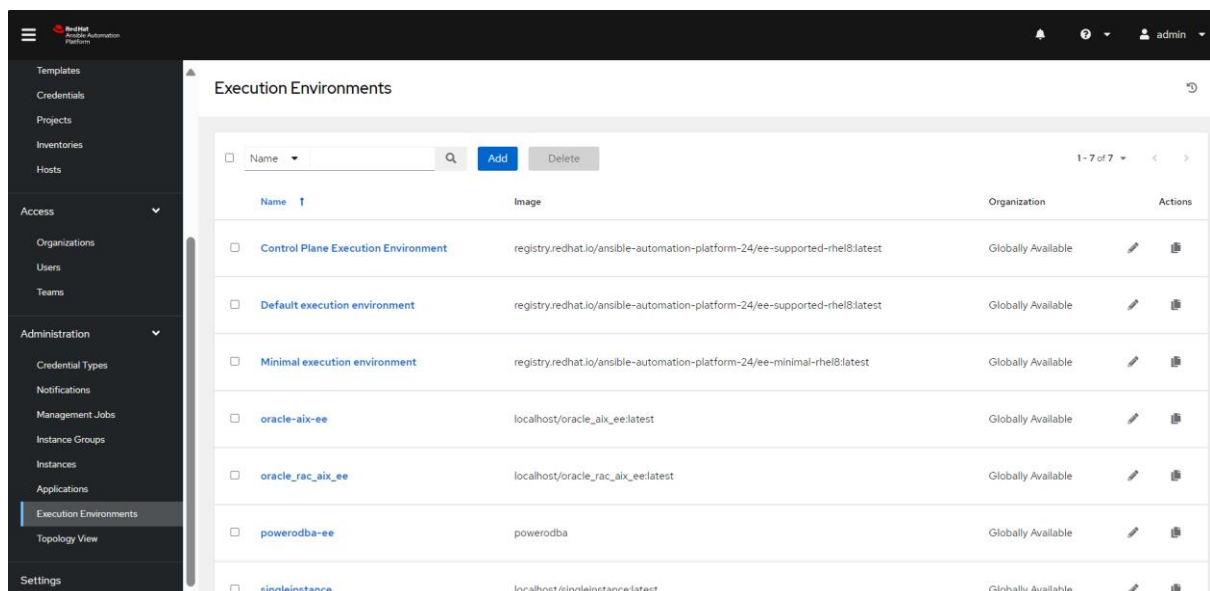
Pull Details: [Provide any additional details for pulling the image]

Once provided, click on "Save" to create the new execution environment.

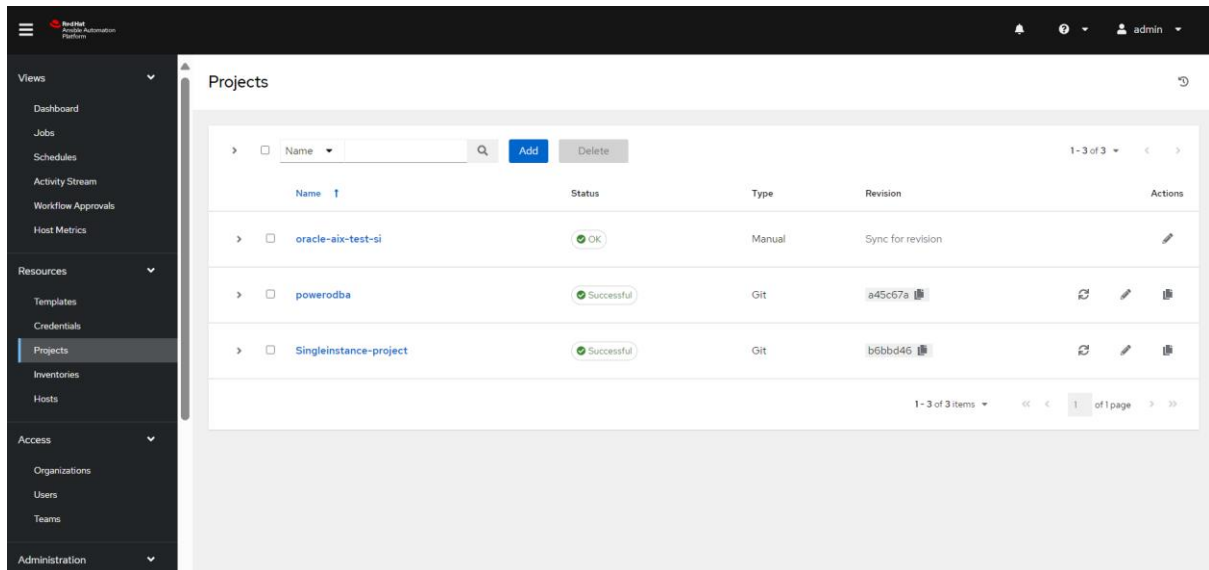
Once saved, we can see the following details to crosscheck.



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



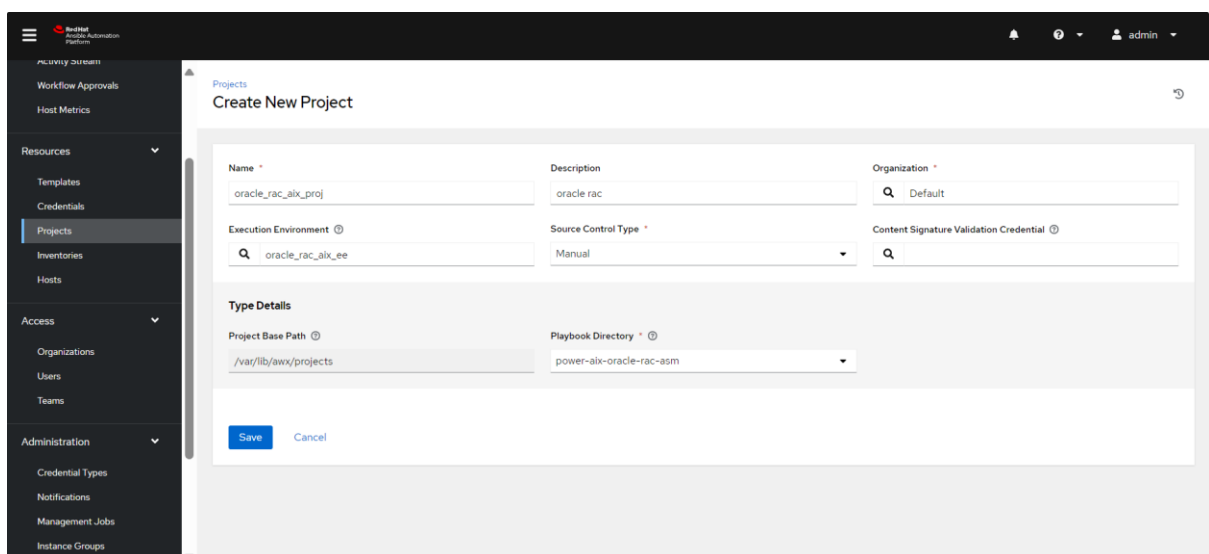
To create Projects, click on the "Projects" option under the Resources dropdown and then click on the "Add" button to create Projects.



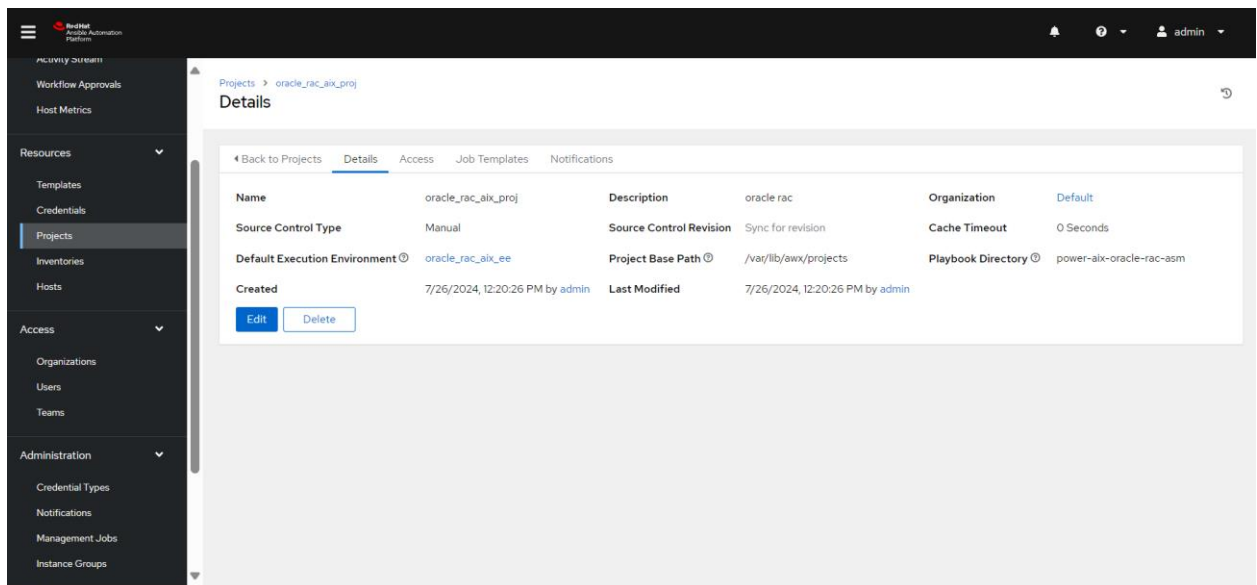
Please provide the following details:

- 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]

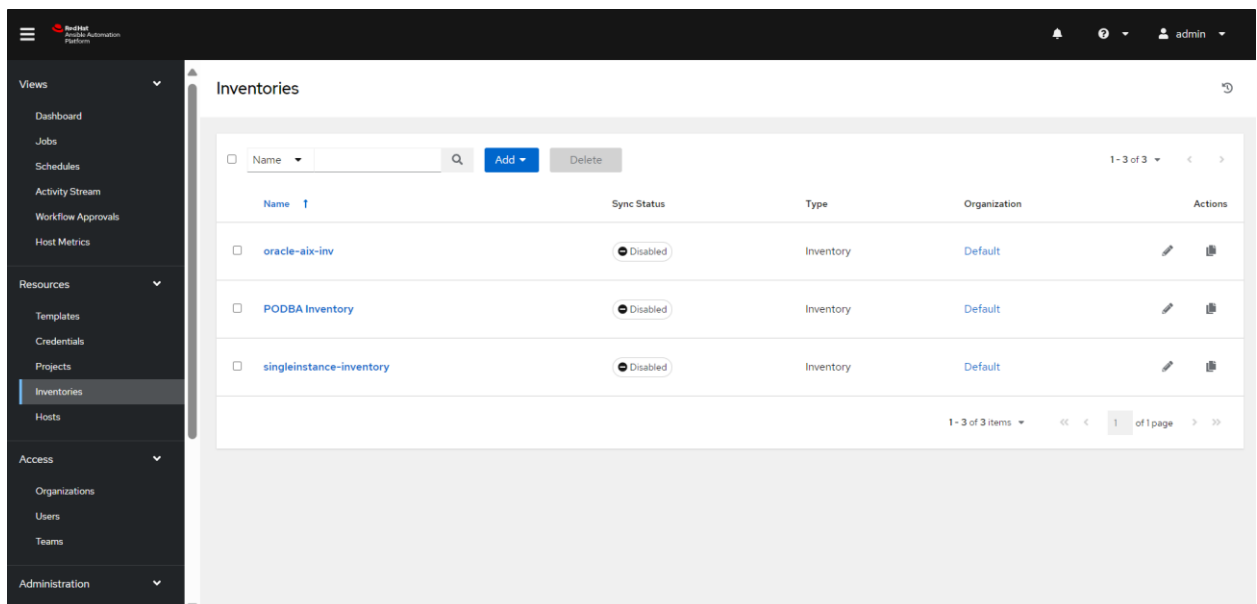
Once provided the details, please click on “Save



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



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".

Create new inventory

Name * Description Organization *

Instance Groups

Labels

Options ☐ Prevent Instance Group Fallback

Variables

1	---

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

Details

Back to Inventories Details Access Groups Hosts Sources Jobs Job Templates

Name Description Type

Organization Total hosts

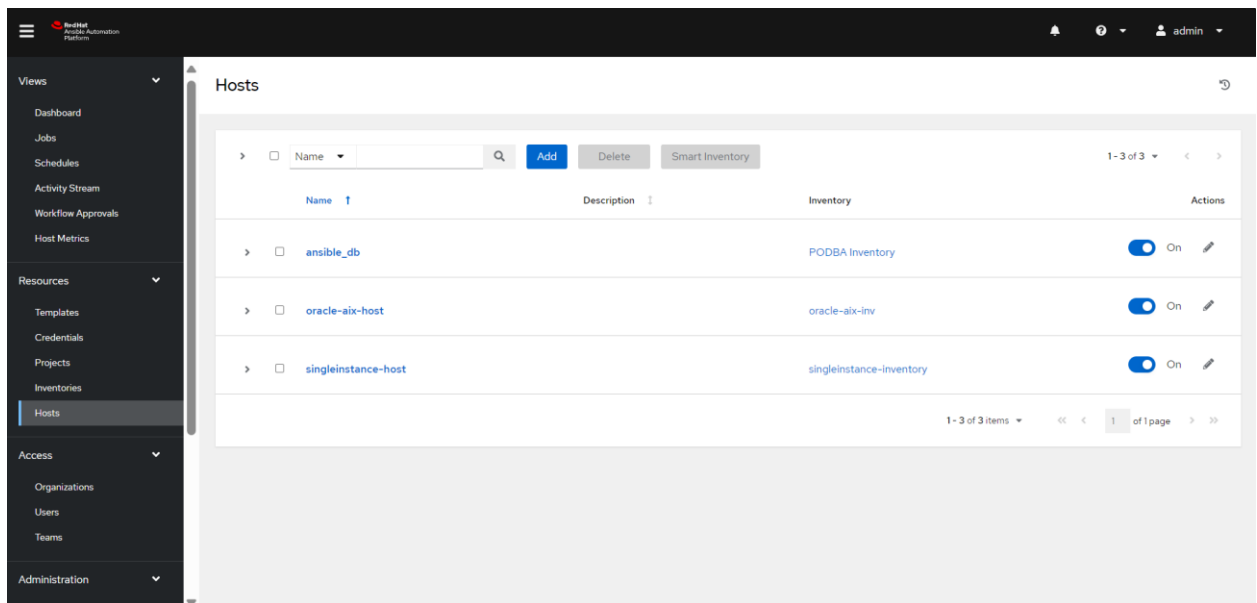
Variables

1	---

Created Last Modified

Create Two Hosts to support RAC

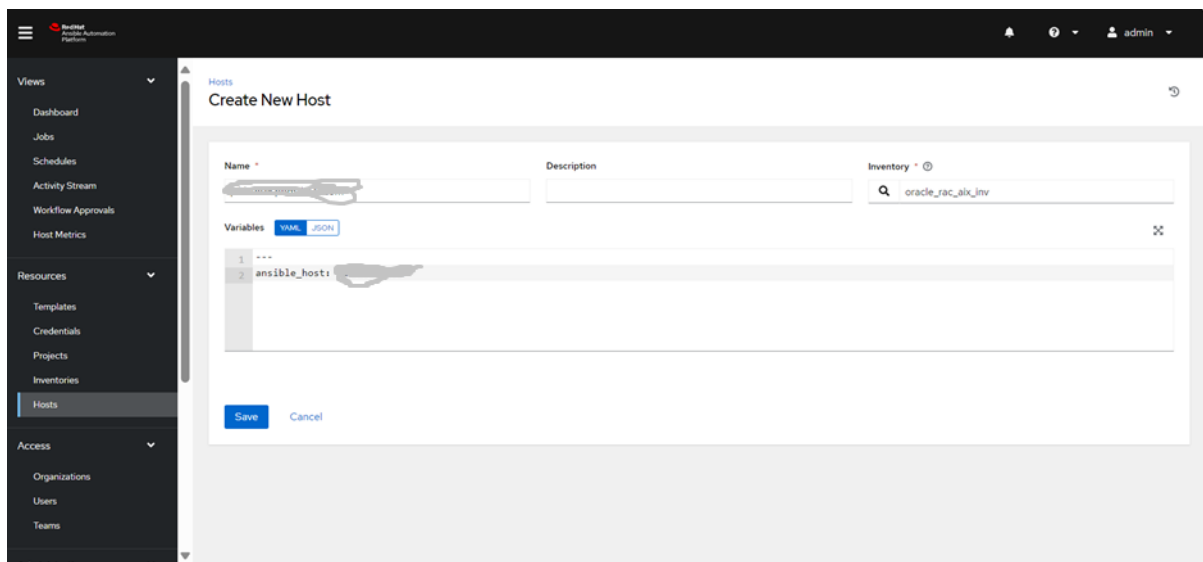
To create First hosts, click on the "Hosts" Option Under the resources dropdown and then click on the "Add" button to Create First Hosts



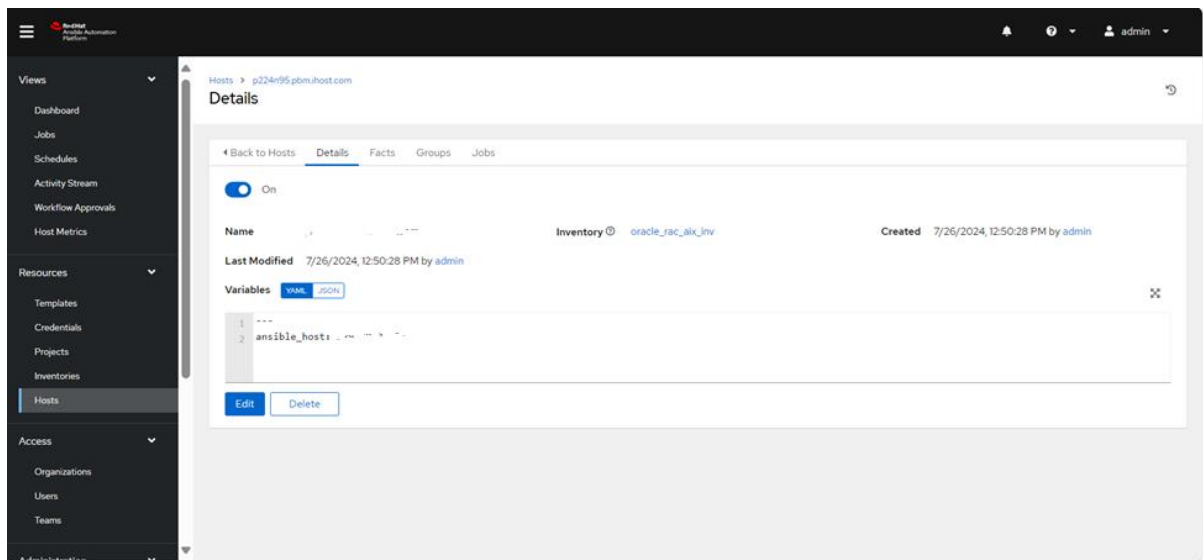
Please provide the following information for the host 1:

- Name: [Enter the name of the hostname of target machine, not with any name]
- Description: [Enter a brief description of your project]
- Inventory: [Select the inventory that was earlier created]
- Variables: ansible-host: <IP Address>

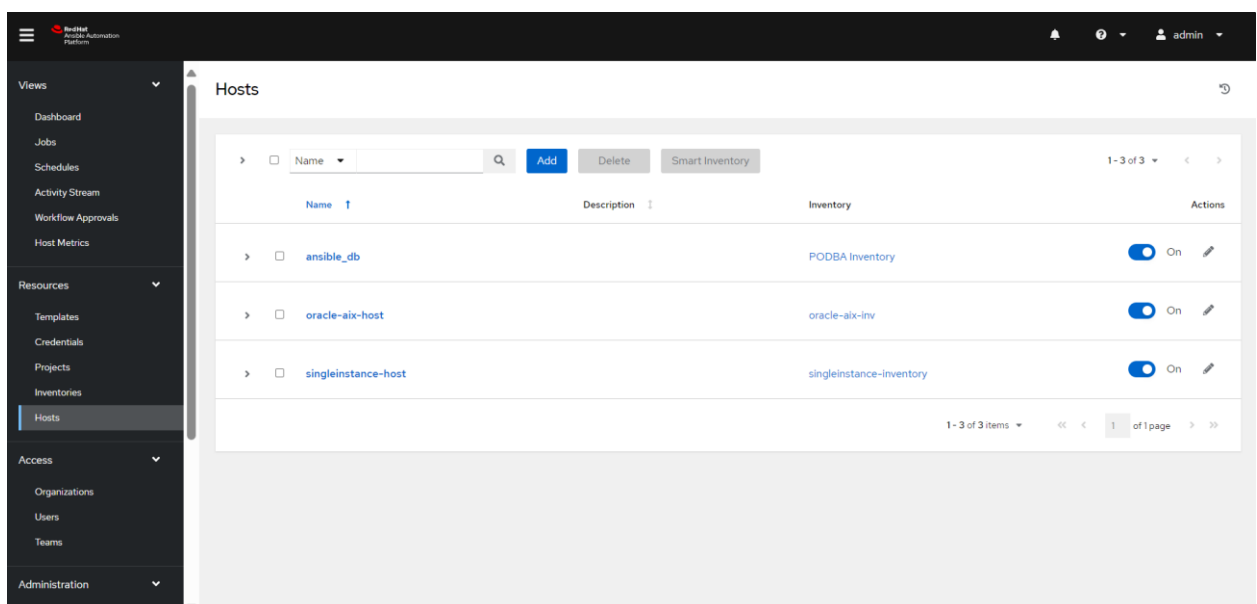
After providing the required information, click on "Save".



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



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



Please provide the following information for the host 2:

- Name: [Enter the name of the hostname of second target machine, not with any name]
- Description: [Enter a brief description of your project]
- Inventory: [Select the inventory that was earlier created]
- Variables: ansible-host

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

Create New Host

Name: Description: Inventory:

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

```

1 ---
2 ansible_host: 1

```

[Save](#) [Cancel](#)

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

Details

Back to Hosts Details Facts Groups Jobs

☒ On

Name: Inventory: Created: 7/26/2024, 12:52:27 PM by admin

Last Modified: 7/26/2024, 12:52:27 PM by admin

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

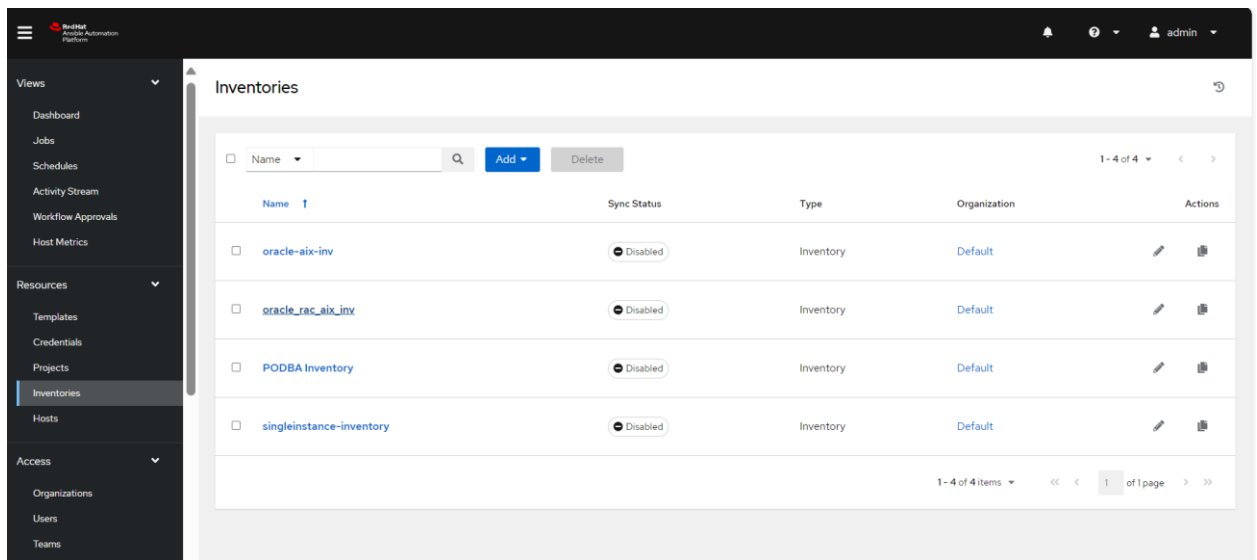
```

1 ---
2 ansible_host: .....

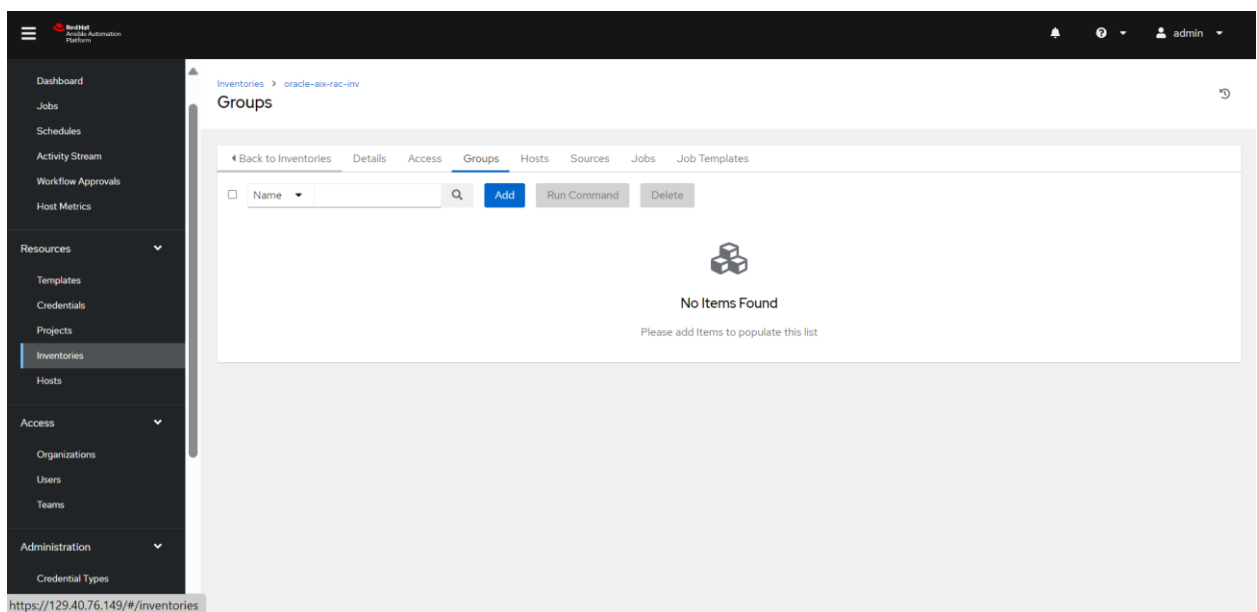
```

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

To create Groups, click on the "Inventories" option under the Resources dropdown and then click on the Inventory which we have created before to this step, in our case the inventory name is "oracle_rac_aix_inv".



On the opened inventory (oracle_rac_aiX_inv) page, click on the Groups tab and Click on “Add” button to create Group.



https://129.40.76.149/#/inventories

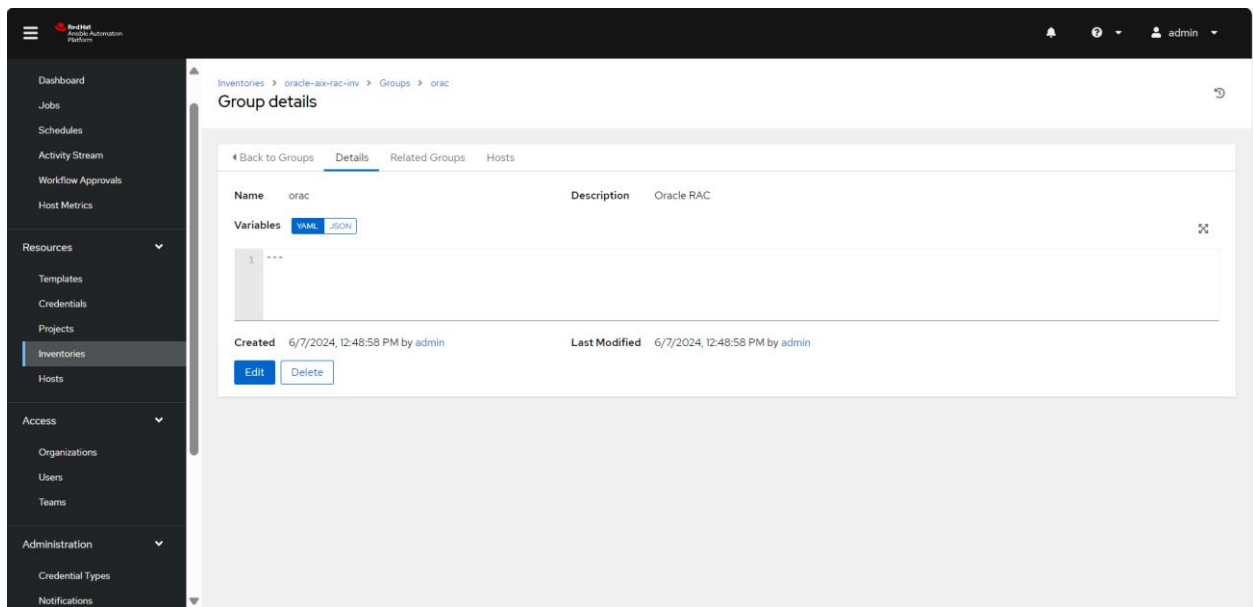
Please provide the following details:

- **Name:** orac
If you decide to use a different name instead of "orac," please update the playbook *'install_and_configure_Oracle_RAC.yml'* file by changing *'- hosts: orac'* to *'- hosts: [new_name]'* and then update the **Name:** field to the new name.
- **Description:** [Enter a brief description of your project]

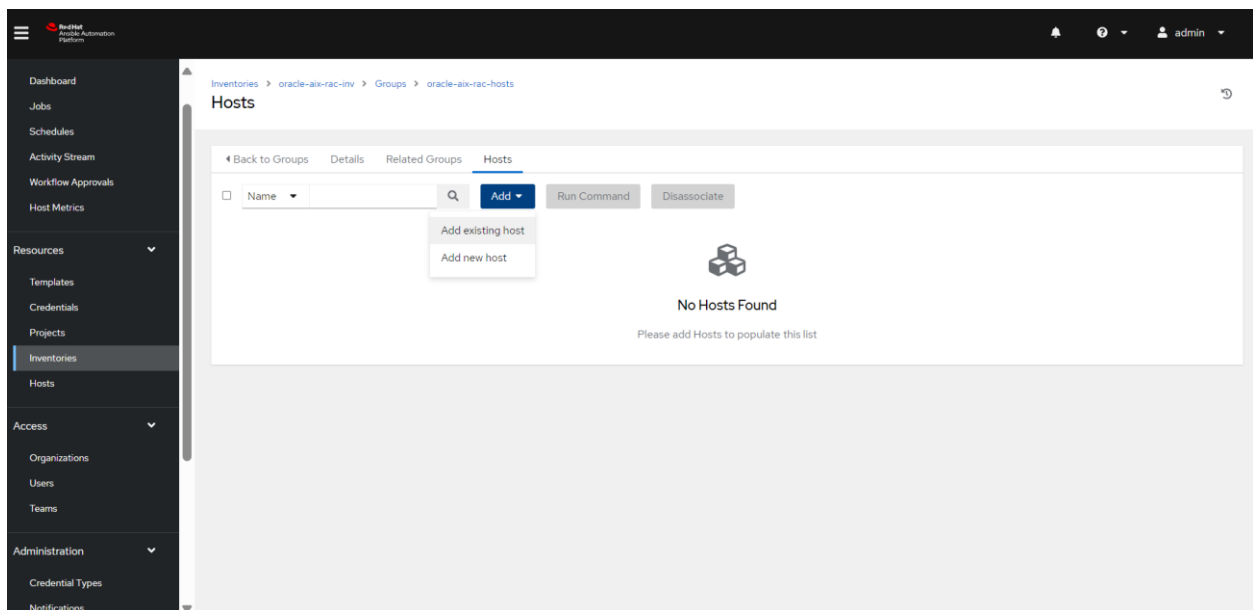
After providing the required information, click on "Save".

The screenshot shows the 'Create new group' form in the Red Hat Ansible Automation Platform. The form is located under the 'Inventories > oracle-aii-rac-inv > Groups' breadcrumb. The 'Name' field is labeled 'Name *' and contains the text 'orac'. The 'Description' field is labeled 'Description' and contains the text 'Oracle RAC'. Below these fields is a 'Variables' section with tabs for 'YAML' and 'JSON'. The 'YAML' tab is selected, and there is a table with one row containing a hyphen ('-'). At the bottom of the form are 'Save' and 'Cancel' buttons. The left sidebar shows the navigation menu with 'Inventories' selected. The top right corner shows the user 'admin'.

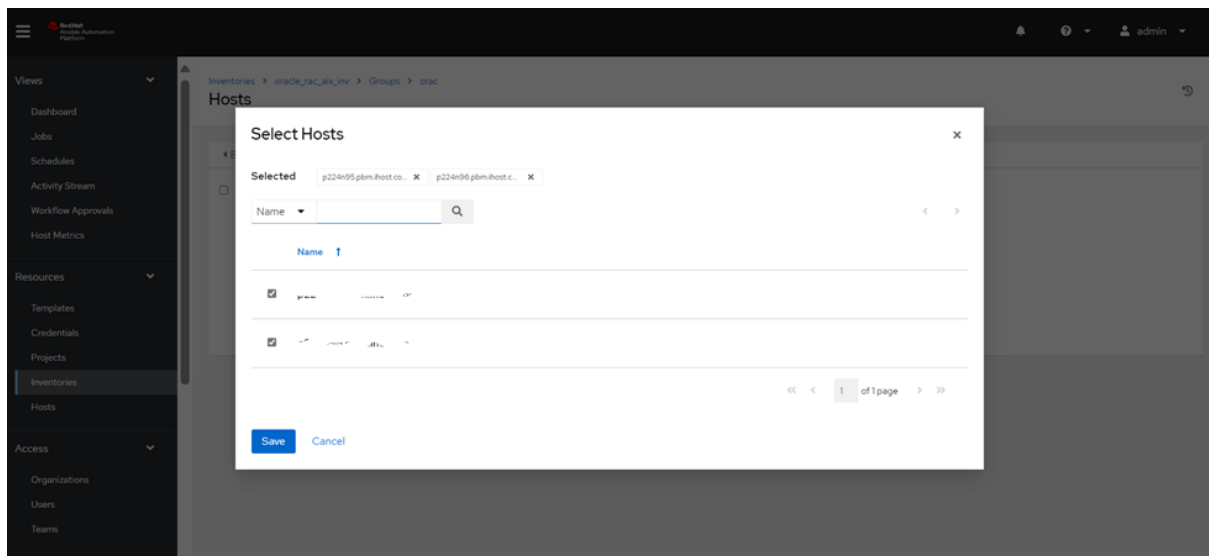
Once saved, we can see the following details to crosscheck.



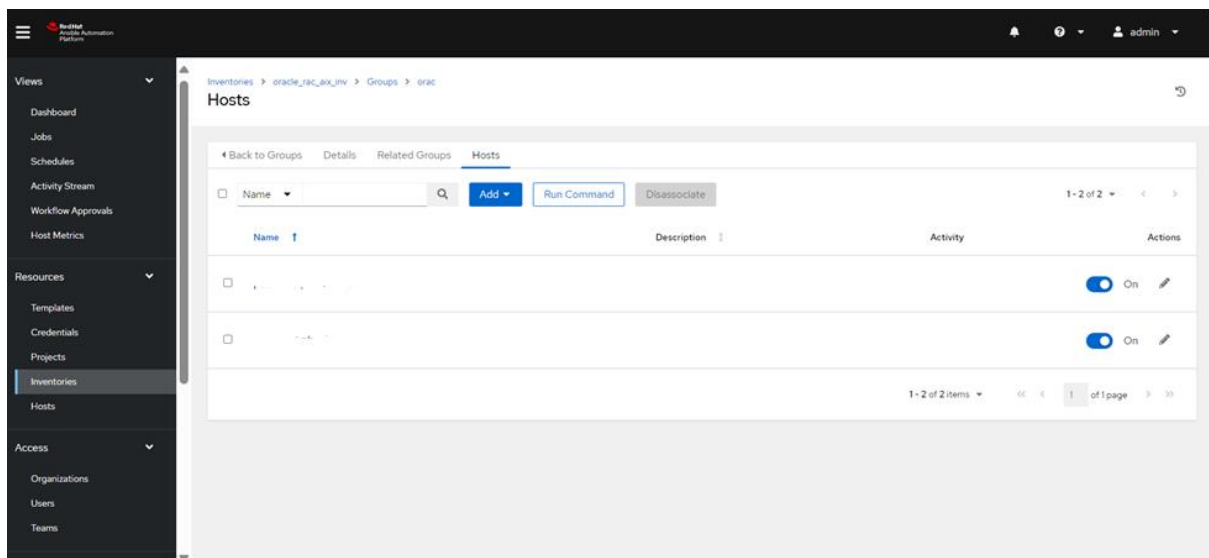
Click on the “Hosts” tab on the Group Details page and Click on “Add” → “Add existing host”



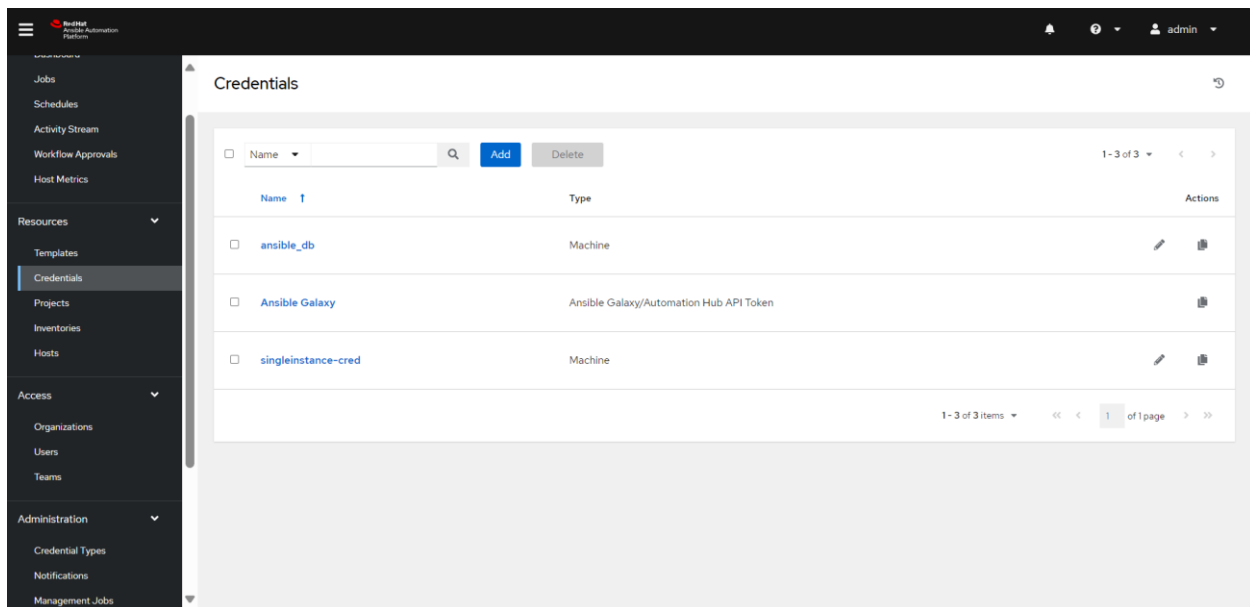
Select the hosts which have created recently and click on “Save”



Now the Hosts have been added to the Group and the Group is added to the inventories.



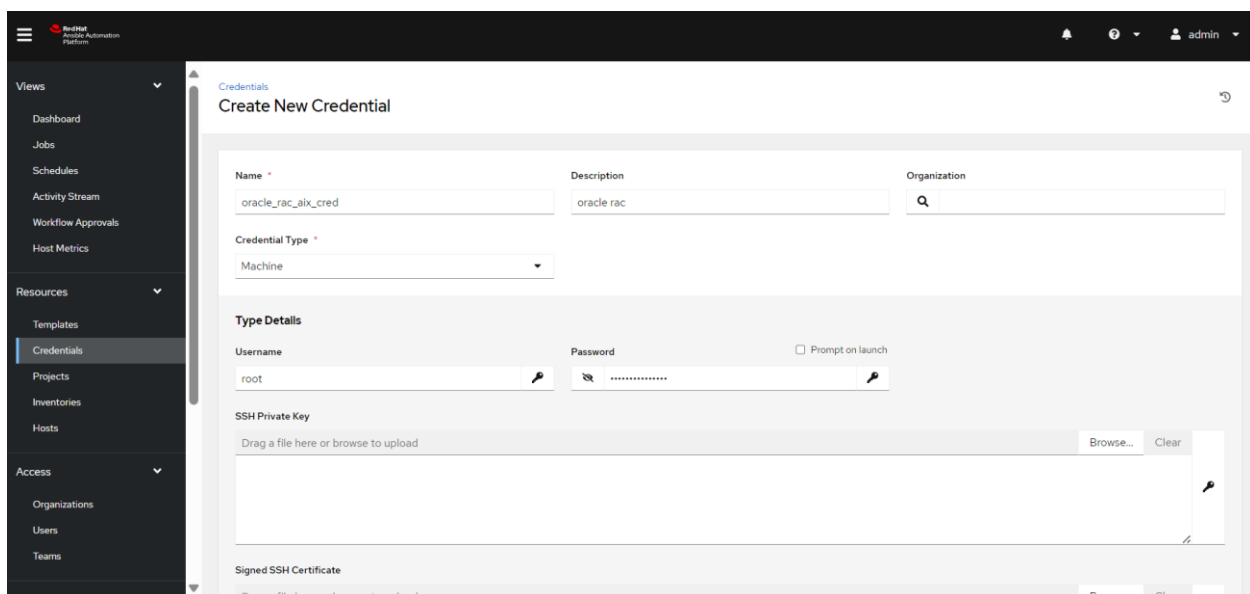
Now, to create Credentials, click on the "Credentials" option under the Resources dropdown and then click on the "Add" button to create Credentials



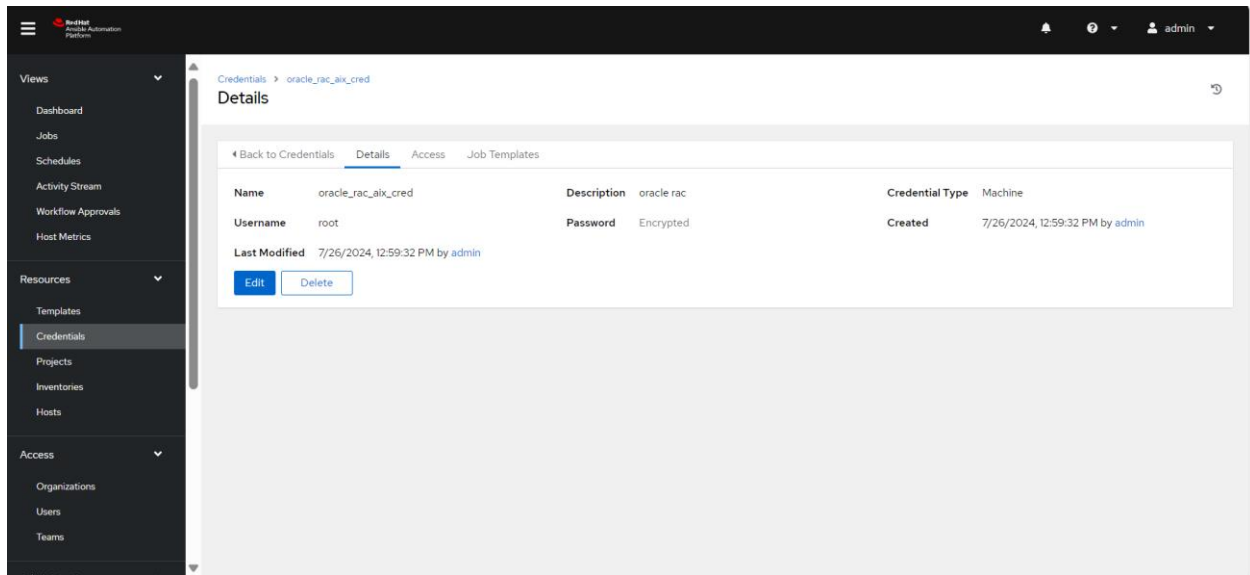
Please provide the following information:

- 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]

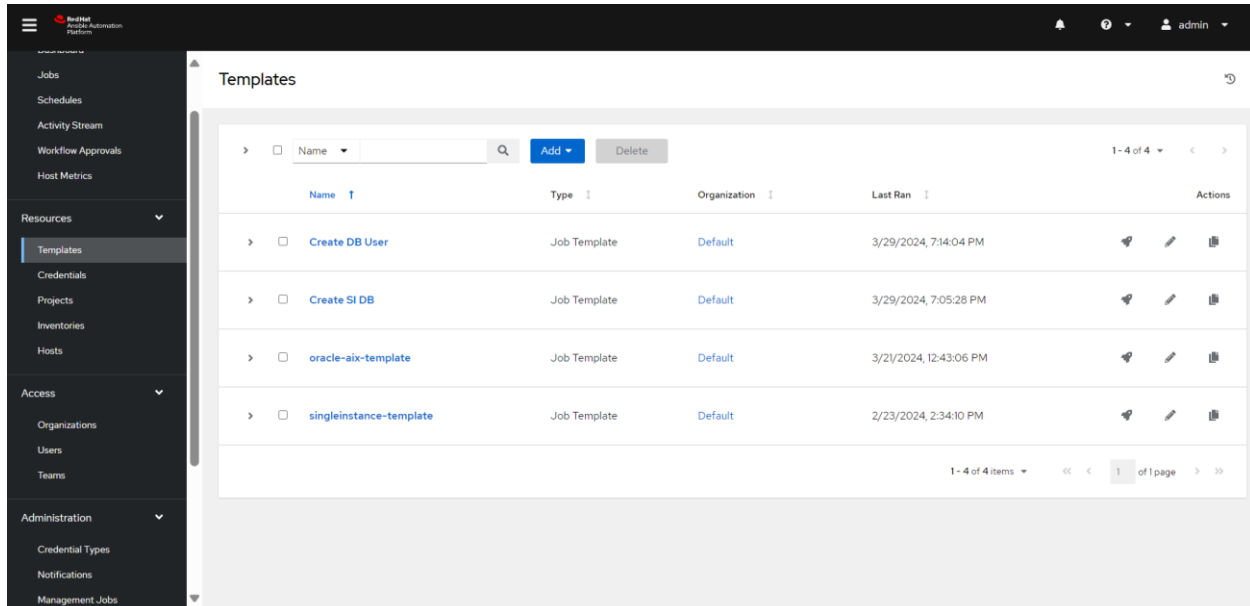
After providing the required information, click on "Save".



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 if you want all the roles ('bootstrap, preconfig, config, install') in the playbook 'install_and_configure_Oracle_RAC.yml' to be executed:

- Name: [Enter the name of your job]
- Description: [Enter a brief description of your job]
- Job Type: Run

- Inventory: [Select the inventory that was earlier created]
- Project: [Select the project that was earlier created]
 - Execution Environment: [Select the execution environment that was earlier created]
- Playbook: [Enter the name of your playbook file]
- Variables: ansible_ssh_user: root

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

Create New Job Template

Name: oracle_rac_aix_template | Description: oracle rac | Job Type: Run | Prompt on launch: ☐

Inventory: oracle_rac_aix_inv | Prompt on launch: ☐ | Project: oracle_rac_aix_proj | Execution Environment: oracle_rac_aix_ee | Prompt on launch: ☐

Playbook: install_and_configure_Oracle_RAC.yml

Credentials: SSH: oracle_rac_aix_c | Prompt on launch: ☐

Labels: | Prompt on launch: ☐

Variables: **YAML** | Prompt on launch: ☐

```

1 ---
2 ansible_ssh_user: root

```

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_rac_aix_template	Description	oracle rac	Job Type	run
Organization	Default	Inventory	oracle_rac_aix_inv	Project	oracle_rac_aix_proj
Execution Environment	oracle_rac_aix_ee	Playbook	install_and_configure_Oracle_RAC.yml	Forks	0
Verbosity	0 (Normal)	Timeout	0	Show Changes	Off
Job Slicing	1	Created	7/26/2024, 1:03:30 PM by admin	Last Modified	7/26/2024, 1:03:30 PM by admin

Credentials: SSH: oracle_rac_aix_c

Variables: **YAML**

```

1 ---
2 ansible_ssh_user: root

```

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

If you want to execute only the bootstrap role, then along with the other details you add "Job Tags"

- Name: [Enter the name of your job]
- Description: [Enter a brief description of your job]
- Job Type: Run
- Inventory: [Select the inventory that was earlier created]
- Project: [Select the project that was earlier created]
- Execution Environment: [Select the execution environment that was earlier created]
- Playbook: [Enter the name of your playbook file]
- Variables: ansible_ssh_user: root
- Job Tags: bootstrap

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

The screenshot shows the 'Create Job' form in the Ansible Automation Platform. The left sidebar contains navigation links for Views, Resources, and Access. The main form area includes fields for Forks (0), Job Slicing (1), Instance Groups, Job Tags (bootstrap), and various options like Privilege Escalation, Provisioning Callbacks, etc. The 'Save' button is at the bottom left.

The screenshot shows the 'Details' page for a job named 'oracle_rac_aix_template'. The page displays various fields including Name, Organization, Execution Environment, Verbosity, Job Slicing, Credentials, Job Tags, and Variables. The 'Variables' field shows a YAML snippet:

```
---
ansible_ssh_user: root
```

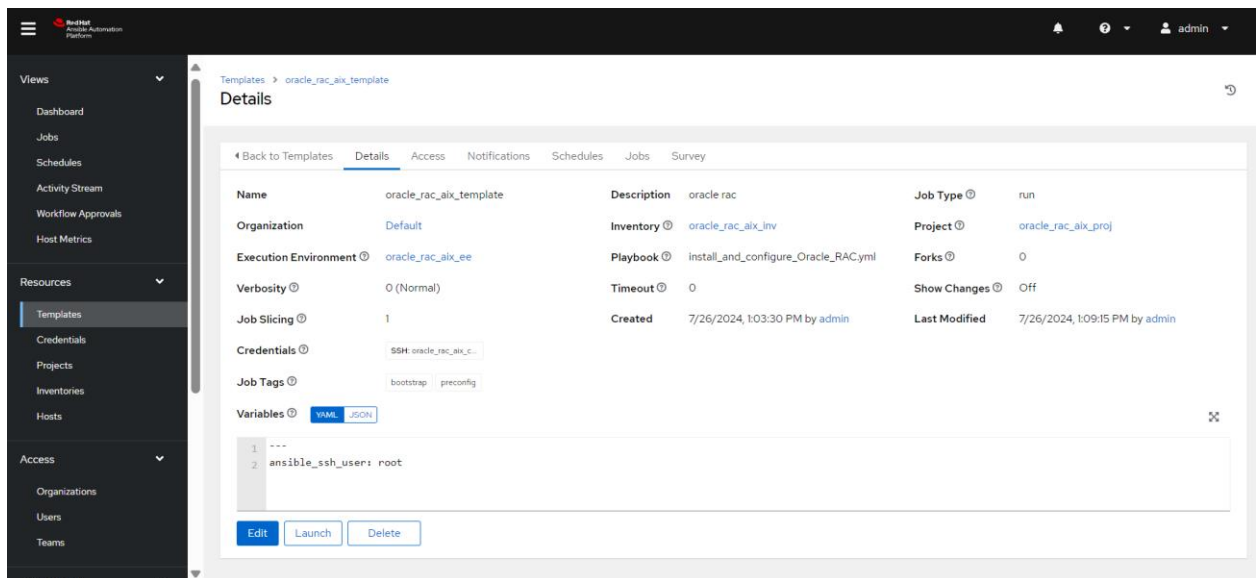
. The 'Save' button is at the bottom left.

If you want to execute only the **'preconfig'** role, then along with the other details, you need to add "Job Tags". Keep in mind that the **'preconfig'** role depends on the **'bootstrap'** role, as the **'bootstrap'** role configures the necessary SSH settings required for the **'preconfig'** role to function properly.

- Name: [Enter the name of your job]
- Description: [Enter a brief description of your job]
- Job Type: Run
- Inventory: [Select the inventory that was earlier created]
- Project: [Select the project that was earlier created]
- Execution Environment: [Select the execution environment that was earlier created]
- Playbook: [Enter the name of your playbook file]
- Variables: ansible_ssh_user: root
- Job Tags: bootstrap, preconfig

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

The screenshot displays the Redhat Ansible Automation Platform (AAP) job configuration page. On the left is a dark sidebar with a menu containing sections: Views (Dashboard, Jobs, Schedules, Activity Stream, Workflow Approvals, Host Metrics), Resources (Templates, Credentials, Projects, Inventories, Hosts), and Access (Organizations, Users, Teams). The 'Templates' option is highlighted. The main content area is a light gray form with the following sections: 'Forks' (input 0, checkbox 'Prompt on launch'), 'Limit' (input, checkbox 'Prompt on launch'), 'Verbosity' (dropdown '0 (Normal)', checkbox 'Prompt on launch'), 'Job Slicing' (input 1, checkbox 'Prompt on launch'), 'Timeout' (input 0, checkbox 'Prompt on launch'), 'Show Changes' (toggle 'Off', checkbox 'Prompt on launch'), 'Instance Groups' (search bar, checkbox 'Prompt on launch'), 'Job Tags' (tags 'bootstrap' and 'preconfig', checkbox 'Prompt on launch'), 'Skip Tags' (input, checkbox 'Prompt on launch'), and 'Options' (checkboxes for 'Privilege Escalation', 'Provisioning Callbacks', 'Enable Webhook', 'Concurrent Jobs', 'Enable Fact Storage', and 'Prevent Instance Group Fallback'). At the bottom are 'Save' and 'Cancel' buttons.



If you want to execute only the **'config'** role, then along with the other details, you need to add "Job Tags". Keep in mind that the **'config'** role depends on the **'bootstrap'** role, as the **'bootstrap'** role configures the necessary SSH settings required for the **'config'** role to function properly.

Note: config role have a dependency on bootstrap role

- Name: [Enter the name of your job]
- Description: [Enter a brief description of your job]
- Job Type: Run
- Inventory: [Select the inventory that was earlier created]
- Project: [Select the project that was earlier created]
- Execution Environment: [Select the execution environment that was earlier created]
- Playbook: [Enter the name of your playbook file]
- Variables: ansible_ssh_user: root
- Job Tags: bootstrap, config

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

Views

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

Resources

- Templates
- Credentials
- Projects
- Inventories
- Hosts

Access

- Organizations
- Users
- Teams

Form Fields:

- Forks**: 0 ☐ Prompt on launch
- Limit**: ☐ Prompt on launch
- Verbosity**: 0 (Normal) ☐ Prompt on launch
- Job Slicing**: 1 ☐ Prompt on launch
- Timeout**: 0 ☐ Prompt on launch
- Show Changes**: ☒ Off ☐ Prompt on launch
- Instance Groups**: ☐ Prompt on launch
- Job Tags**: bootstrap config ☐ Prompt on launch
- Skip Tags**: ☐ Prompt on launch
- Options**:
 - ☐ Privilege Escalation
 - ☐ Provisioning Callbacks
 - ☐ Enable Webhook
 - ☐ Concurrent Jobs
 - ☐ Enable Fact Storage
 - ☐ Prevent Instance Group Fallback

Buttons: Save Cancel

Views

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

Resources

- Templates
- Credentials
- Projects
- Inventories
- Hosts

Access

- Organizations
- Users
- Teams

Template Details: oracle_rac_aix_template

Back to Templates		Details	Access	Notifications	Schedules	Jobs	Survey
Name	oracle_rac_aix_template	Description	oracle rac	Job Type	run		
Organization	Default	Inventory	oracle_rac_aix_inv	Project	oracle_rac_aix_proj		
Execution Environment	oracle_rac_aix_ee	Playbook	install_and_configure_Oracle_RAC.yml	Forks	0		
Verbosity	0 (Normal)	Timeout	0	Show Changes	Off		
Job Slicing	1	Created	7/26/2024, 1:03:30 PM by admin	Last Modified	7/26/2024, 1:10:57 PM by admin		
Credentials	SSH: oracle_rac_aix_c...						
Job Tags	bootstrap config						
Variables	YAML JSON						
<pre> 1 --- 2 ansible_ssh_user: root </pre>							

Buttons: Edit Launch Delete

If you want to execute only the **'install'** role, then along with the other details, you need to add "Job Tags". Keep in mind that the **'install'** role depends on the **'bootstrap'** role, as the **'bootstrap'** role configures the necessary SSH settings required for the **'install'** role to function properly.

- Name: [Enter the name of your job]
- Description: [Enter a brief description of your job]
- Job Type: Run
- Inventory: [Select the inventory that was earlier created]
- Project: [Select the project that was earlier created]
- Execution Environment: [Select the execution environment that was earlier created]
- Playbook: [Enter the name of your playbook file]

- Variables: ansible_ssh_user: root
- Job Tags: bootstrap, install

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

The screenshot shows the 'Create Template' form in the Red Hat Ansible Automation Platform. The left sidebar contains navigation links for Views, Resources, and Access. The main form area includes fields for Forks (0), Limit (empty), Verbose (0 (Normal)), Job Slicing (1), Timeout (0), Show Changes (Off), Instance Groups (empty), Job Tags (bootstrap, install), Skip Tags (empty), and Options (Privilege Escalation, Provisioning Callbacks, Enable Webhook, Concurrent Jobs, Enable Fact Storage, Prevent Instance Group Fallback). At the bottom are 'Save' and 'Cancel' buttons.

The screenshot shows the 'Template Details' page for the 'oracle_rac_aix_template'. The page has tabs for Back to Templates, Details, Access, Notifications, Schedules, Jobs, and Survey. The Details tab is active, showing a table with fields: Name (oracle_rac_aix_template), Description (oracle rac), Job Type (run), Organization (Default), Inventory (oracle_rac_aix_inv), Project (oracle_rac_aix_proj), Execution Environment (oracle_rac_aix_ee), Playbook (install_and_configure_Oracle_RAC.yml), Forks (0), Verbose (0 (Normal)), Timeout (0), Show Changes (Off), Job Slicing (1), Created (7/26/2024, 1:03:30 PM by admin), Last Modified (7/26/2024, 1:12:40 PM by admin), Credentials (SSH oracle_rac_aix_c...), Job Tags (bootstrap, install), and Variables (YAML, JSON). The Variables field is expanded, showing a code editor with the content: '---\nansible_ssh_user: root'. At the bottom are 'Edit', 'Launch', and 'Delete' buttons.

Appendix

- **Programming languages used**

Apart from YAML and Jinja2, many tasks are implemented in Korn Shell 93 and Perl.

- **Customization/enhancement requests**

The files under each role directory have been tested. Unless additional functionalities/features are desired beyond this release provides, making changes should proceed with caution. Good understanding of the task flow and dependency is important to ensure successfully outcome.

Alternatively, it's recommended to submit enhancement requests.

- **Limitations**

- The Oracle Grid HOME is on a local JFS2 filesystem created within an AIX volume group
The Oracle database HOME is created on shared disks behind an ASM's ACFS disk group and also on a local JFS2 filesystem.
- NFS is needed to stage the oracle software
- Passwords in plain text are stored in in vars.yml and in the scripts. Some templated scripts contain passwords are pushed out to the RAC nodes. Security measures should be taken such as limit the access to the Ansible controller and the RAC nodes prior and during the installation and change the passwords as soon as the installation is successful.
- Proxy environment is not supported.
- Grid Naming Service (GNS) virtual IP address is not supported.
- OCR and voting disks are supported on ASM.

- **Openssl check**

All openssl dependent packages updated after Aug 5, 2022 require openssl 1.1. The bootstrap and config roles which use yum/dnf will get fail if you are trying to install the openssl dependent packages/rpms when openssl version is < 1.1.

Openssl fileset needs to be upgraded manually, refer to below url for steps

<https://www.ibm.com/support/pages/node/6833478>

<https://www.ibm.com/support/pages/node/720655>

```
# lsipp -l|grep openssl
openssl.base      1.0.2.2104 COMMITTED Open Secure Socket Layer
openssl.license   1.0.2.2104 COMMITTED Open Secure Socket License
openssl.man.en_US 1.0.2.2104 COMMITTED Open Secure Socket Layer
openssl.base      1.0.2.2104 COMMITTED Open Secure Socket Layer
```

- **Disk headers and PVIDs**

Disks may have AIX volume group, Oracle ASM diskgroup headers, and PVIDs left on the disks by a previous project. When creating a volume group or an ASM diskgroup, the PVIDs and/or headers cause the volume group or ASM diskgroup creation to fail. If it is certain the disks can be reused, clear the header on the disks before starting the playbook like this:

```
dd if=/dev/zero of=/dev/hdiskX bs=1024k count=100
```

To determine a disk has an ASM diskgroup header, use `lquerypv -h /dev/hdiskX`

<https://www.ibm.com/support/pages/do-not-mix-oracle-asm-disks-lvm>

Example showing ASM header:

```
# lquerypv -h /dev/hdisk23
```

```
00000000 00820101 00000000 80000000 BF477E44 |.....G~D|
00000010 00000000 00000000 00000000 00000000 |.....|
00000020 4F52434C 4449534B 00000000 00000000 |ORCLDISK.....|
00000030 00000000 00000000 00000000 00000000 |.....|
00000040 13000000 00000103 4F435256 4F54455F |.....OCRVOTE_|
00000050 30303030 00000000 00000000 00000000 |0000.....|
00000060 00000000 00000000 4F435256 4F544500 |.....OCRVOTE_|
00000070 00000000 00000000 00000000 00000000 |.....|
00000080 00000000 00000000 4F435256 4F54455F |.....OCRVOTE_|
00000090 30303030 00000000 00000000 00000000 |0000.....|
000000A0 00000000 00000000 00000000 00000000 |.....|
000000B0 00000000 00000000 00000000 00000000 |.....|
000000C0 00000000 00000000 01F95E4D 8CACD800 |.....^M....|
000000D0 01F95E4D 9DE2F000 02001000 00400000 |..^M.....@..|
000000E0 0006EE80 00001400 00000003 00000001 |.....|
000000F0 00000002 0000000A 00000000 00000000 |.....|
```

To determine if a disk has an on-disk PVID, use `lqueryvg -Ptp hdiskX`

<https://www.ibm.com/support/pages/resolving-missing-or-removed-disks-aix-lvm>

If PVID exists, the exit code is 0, otherwise it is 1.

To determine if a disk has a Volume Group Descriptor Area (VGDA), use `readvgda -t hdiskX`.

<https://www.ibm.com/support/pages/how-determine-volume-group-physical-partition-size-readvgda>

If VGDA exists, the exit code is 0, otherwise it is 1.

Alternatively, if disks can be reused for sure, the `vars.yml` config `asmdisks.diskgroups` and `config.ora_vg` has options “`clear_pvids`” and “`zero_disks`” which free the user from executing the extra commands to clear the PVIDs, AIX volume group headers, and ASM diskgroup headers.

Trademarks

IBM® is a registered trademark of IBM Corporation in the United States and other countries

POWER9® is a registered trademark of IBM Corporation in the United States and other countries

POWER8® is a registered trademark of IBM Corporation in the United States and other countries

POWER® is a registered trademark of IBM Corporation in the United States and other countries

AIX® is a registered trademark of IBM Corporation in the United States and other countries

Oracle® is a registered trademark of Oracle Corporation in the United States and other countries

Ansible® is a registered trademark of Red Hat® Inc. in the United States and other countries

Red Hat® is a registered trademark of Red Hat Inc. in the United States and other countries