

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

Description

This playbook assumes the following:

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

To get started with Ansible refer

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

To get started with Oracle Database on AIX refer

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

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

To get started with AIX refer

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

System Configuration

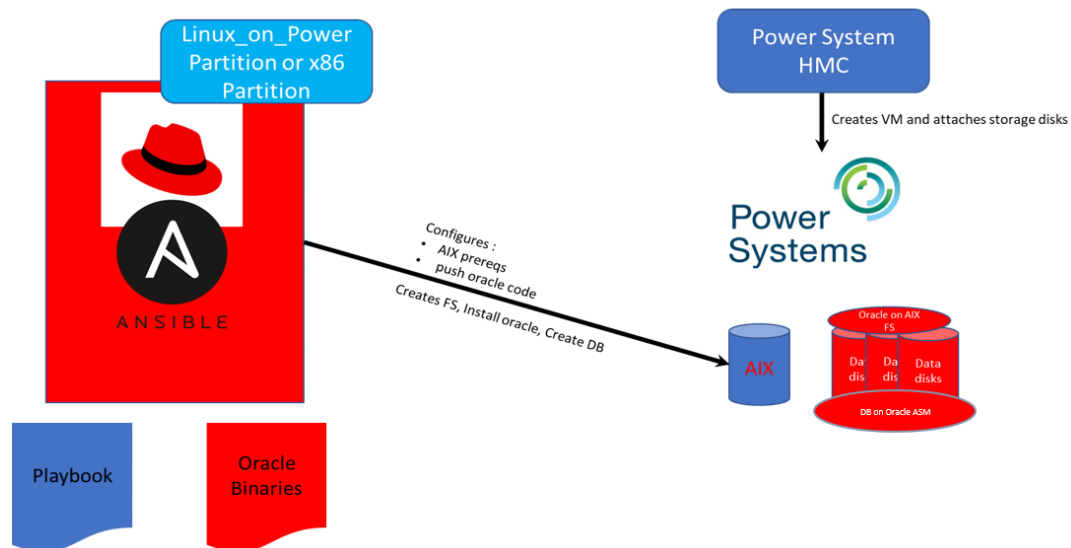


Figure. System Topology

Below is the system configuration that we have used for testing

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

a) Linux_on_Power server :

Operating System : RHEL 8.2

Ansible Engine Version : 2.10.2

For Ansible Engine prerequisites refer to below link

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

b) AIX server :

Operating System : AIX72TL4SP01

Oracle DB Version : 19.3.0.0.0

CPUs : 4

RAM : 64GB

Storage Disks : 2X40GB (one rootvg and another for oracle DB)

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 Followed

1. Install Ansible Engine on your preferred operation system. We have installed and tested Ansible Engine on x86 server and Linux-on-Power server

Refer : https://docs.ansible.com/ansible/latest/installation_guide/intro_installation.html

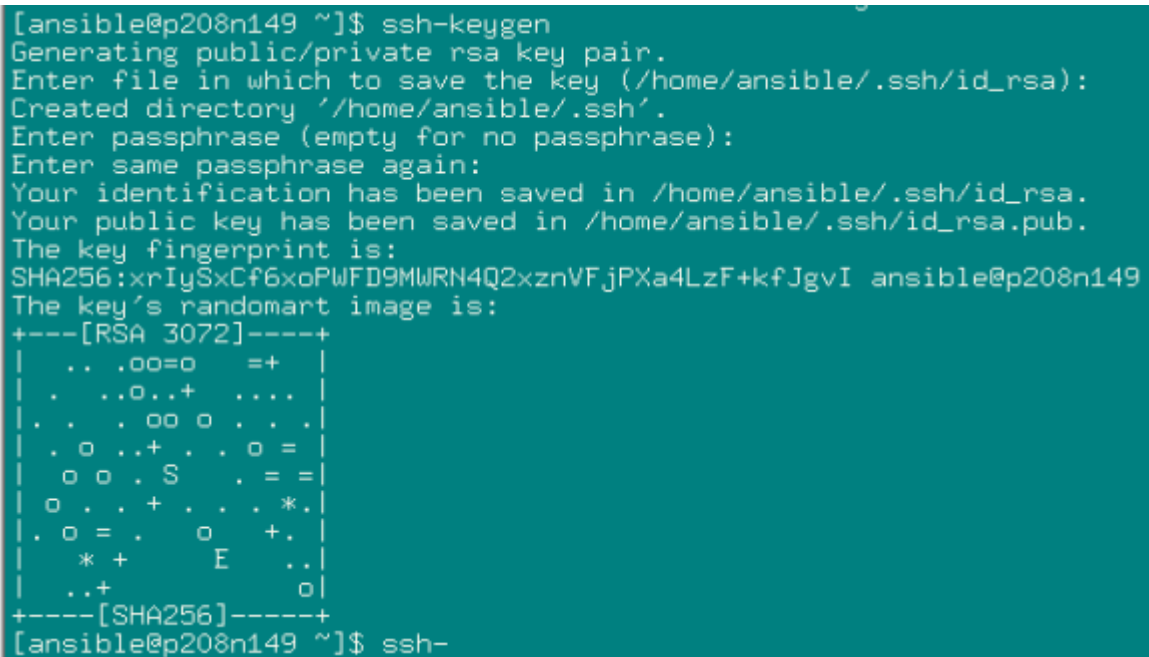
We have created a user “ansible” on LoP server and considered /home/ansible as working Directory.

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

If this is a first time using ssh, then you probably haven’t created your ssh keys. To check go to ~/.ssh and see if id_rsa file exists. If not you must create the ssh keys.

To create the ssh keys, run the following:

```
➤ ssh-keygen
```



```
[ansible@p208n149 ~]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ansible/.ssh/id_rsa):
Created directory '/home/ansible/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ansible/.ssh/id_rsa.
Your public key has been saved in /home/ansible/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:xrIySxCf6xoPWFD9MWRN4Q2xznVFjPXa4LzF+kfJgvI ansible@p208n149
The key's randomart image is:
+---[RSA 3072]-----+
|  .. .oo=o  =+  |
|  .  ..o..+  ....|
|  .  .  oo o  .  .|
|  . o ..+ .  . o =|
|  o o . S  .  = =|
|  o . . + .  .  *|
|  . o = .  o  +.  |
|  * +      E  ..  |
|  ..+          o  |
+-----[SHA256]-----+
[ansible@p208n149 ~]$ ssh-
```

Next copy the keys to the managed host.

```
➤ ssh-copy-id root@<managed_host>
```

Eg: ssh-copy-id root@p227n241

```
[ansible@p208n149 ~]$ ssh-copy-id root@p227n241
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/ansible/.ssh/id_rsa.pub"
The authenticity of host 'p227n241 (129.40.76.241)' can't be established.
ECDSA key fingerprint is SHA256:ykIrGsVK+rc17RPwujt6WhZ1mITZsjx90NfK+MVReHM.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s),
to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed --
if you are prompted now it is to install the new keys
root@p227n241's password:
sh: test: argument expected

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'root@p227n241'"
and check to make sure that only the key(s) you wanted were added.
```

As per the last step instructions from the previous step, test that you can access the managed host:

```
➤ ssh 'root@p227n241'
```

```
> ssh 'root@p227n241'
Last login: Tue Mar 16 17:05:45 2021 on /dev/pts/1 from 129.40.76.149
*****
*
* Welcome to AIX Version 7.2!
*
* Please see the README file in /usr/lpp/bos for information pertinent to
* this release of the AIX Operating System.
*
*
*****
# exit
Connection to p227n241 closed.
> █
```

3. Preparing to run the oracle playbook

- Download the Oracle AIX playbook ansible collection from ansible galaxy or github.
https://galaxy.ansible.com/ibm/power_aix_oracle
<https://github.com/IBM/ansible-power-aix-oracle>

To download from github you can follow below steps

```
$ cd <working directory>

$ git clone https://github.com/IBM/ansible-power-aix-oracle.git

$ cd ansible-power-aix-oracle
```

Download the power_aix_oracle tarball from ansible-galaxy and extract it to some directory for offline use **or** you can run the ansible-galaxy installation command:

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

The above command will install power_aix_oracle collection at location ~/.ansible/collections

For more information regarding ansible collections refer below link

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

Before running the playbook you should do

- a) This collection uses ibm.power_aix collection modules like filesystem, devices, lvgl and mount. Install latest version (1.4.0 or later) from galaxy
"ansible-galaxy collection install ibm.power_aix"
- b) Download the Oracle 19c software from OTN or oracle edelivery site
<https://edelivery.oracle.com/osdc/faces/SoftwareDelivery>
<https://www.oracle.com/database/technologies/oracle19c-aix-193000-downloads.html>
- c) Added Boolean variable **grid_asm_flag** for Grid installation. Value "true" indicates the DB will get created on ASM disks and value "false" indicates the DB will get created on JFS which is default option.
- d) Modify the Oracle Binary location path variable "oracledbaix19c" in file "vars/oracle_params.yml"
- e) 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'
- f) Check other Oracle related parameters in file "vars/oracle_params.yml", modify it based on your need
- g) Based on your environment update resolv.conf and netnsvc.conf files at
"roles/preconfig/files/"
- h) There should be atleast one free disk available other than rootvg for Oracle DB Installation and test database creation on JFS filesystem. Make sure disk header information is clean. You can check the header information using "lquerypv -h /dev/hdiskX". These free disks are used for staging oracle software binary and oracle datafiles. Minimum 40GB disk storage is needed for running this Oracle playbook.
- i) The rootvg disk should be atleast 30GB, we will be using /tmp for ansible remote location(~8GB) and assumes paging device part of rootvg

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

The collection contains below three roles

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

Inside power_aix_oracle collection go to "playbooks" directory

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

```
$ cat ansible.cfg
[defaults]
inventory = ./inventory
# interpreter_python = /usr/opt/freeware/bin/python3 # AIX 7.3 supports python3
interpreter_python = /usr/bin/python
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
p227n241
```

4. Execute playbook using below command

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

Note: AIX 7.3 uses dnf for installing the packages by default it will be installed at /opt/freeware/bin. If not, use “demo_bootstrap_dnf.yml” from ibm.power_aix collection for configuring dnf and python3

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

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

$ ansible-playbook demo_play_aix_oracle.yml
```

You can also run each role separately using ansible tags

To run only preconfig tasks

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

If you want to run preconfig and oracle_install tasks

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

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

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

Playbook Roles

As discussed earlier this collection has three roles

1) preconfig :

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

2) oracle_install:

- Detecting oracle version to install
- Create Oracle groups and user
- Creating volume group for ORACLE_HOME
- Creating and mounting filesystem for ORACLE_HOME
- Creating oracle installation directories
- If grid option selected install Standalone Grid Software
- Updating .profile file with Oracle env details
- Generating oracle response file and install Oracle DB Software
- Run root scripts

3) oracle_createdb:

- Check /etc/oratab file for DB existence

- If grid option selected create database on ASM storage
- For JFS DB, create VG and mount filesystems
- Generate Database creation template file
- Generate database creation script
- Creating database
- For JFS DB, Creating and configuring oracle listener
- Check Oracle PMON background process status

Executing Oracle collection using Ansible Automation Platform 2 (AAP2)

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

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

At first using “ansible-builder” create Containerfile or Dockerfile and then build the Container Image

For more info regarding ansible-builder refer to below

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

```
$ cat execution-environment.yml
---
version: 1

build_arg_defaults:
  EE_BASE_IMAGE: 'quay.io/ansible/ansible-runner:latest'

dependencies:
  galaxy: requirements.yml

additional_build_steps:
  append:
    - RUN mkdir -p /zips/oraclesw19c/
    - COPY oraclesw19c/AIX.PPC64_193000_db_home.zip /zips/oraclesw19c/
    - COPY oraclesw19c/AIX.PPC64_193000_grid_home.zip /zips/oraclesw19c/
```

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


After creating execution-environment.yml file use ansible-builder to create Dockerfile or Containerfile

```
$ ansible-builder create -f execution-environment.yml
Complete! The build context can be found at: /var/lib/awx/test/oracle/context
```

Displaying the content of container or docker file

```
$ cd context
$ cat Containerfile
ARG EE_BASE_IMAGE=quay.io/ansible/ansible-runner:latest
ARG EE_BUILDER_IMAGE=quay.io/ansible/ansible-builder:latest

FROM $EE_BASE_IMAGE as galaxy
ARG ANSIBLE_GALAXY_CLI_COLLECTION_OPTS=
USER root

ADD _build /build
WORKDIR /build

RUN ansible-galaxy role install -r requirements.yml --roles-path /usr/share/ansible/roles
RUN ansible-galaxy collection install $ANSIBLE_GALAXY_CLI_COLLECTION_OPTS -r
requirements.yml --collections-path /usr/share/ansible/collections

FROM $EE_BUILDER_IMAGE as builder

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

RUN ansible-builder introspect --sanitize --write-bindep=/tmp/src/bindep.txt --write-
pip=/tmp/src/requirements.txt
RUN assemble

FROM $EE_BASE_IMAGE
USER root

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

COPY --from=builder /output/ /output/
RUN /output/install-from-bindep && rm -rf /output/wheels
RUN mkdir -p /zips/oraclesw19c/
COPY oraclesw19c/AIX.PPC64_193000_db_home.zip /zips/oraclesw19c/
COPY oraclesw19c/AIX.PPC64_193000_grid_home.zip /zips/oraclesw19c/
```

Next manually copy oracle binary software directory to context directory. In next release we will have a nfs option for staging the oracle binary files, this will eliminate the copy of binary files to container.

```
$ cp /zips/oraclesw19c /var/lib/awx/test/oracle/context
```

Using ansible-builder build the execution environment

```
$ cd /var/lib/awx/test/oracle

$ ansible-builder build -t oracle_aix_ee
Running command:
  podman build -f context/Containerfile -t oracle_aix_ee context
Complete! The build context can be found at: /var/lib/awx/test/oracle/context

$ podman images
REPOSITORY          TAG       IMAGE ID   CREATED    SIZE
localhost/oracle_aix_ee  latest   8029c770954b About a minute ago 7.91 GB
<none>              <none>   037f938762cb 2 minutes ago 914 MB
<none>              <none>   61d13beee50b 3 minutes ago 835 MB
quay.io/ansible/ansible-runner latest    40014730d1b7 18 hours ago 833 MB
quay.io/ansible/ansible-builder latest    b0348faa7f41 8 weeks ago 779 MB
```

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

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

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

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

Executing the Playbook from Ansible Controller (Tower) using execution environment

Login to ansible controller and ADD the newly created execution environment

The screenshot shows the 'Edit details' page for an execution environment named 'oracle-ai-x-ee'. The 'Name' field is 'oracle-ai-x-ee' and the 'Image' field is 'localhost/oracle_ai_x_ee:latest'. The 'Description' field is empty. The 'Organization' field is set to 'Default'. The 'Registry credential' field is empty. The 'Pull' dropdown is set to 'Only pull the image if not present before running...'. The 'Save' button is highlighted in blue.

Click save and the container image will get listed in Execution Environments

The screenshot shows the 'Execution Environments' list. The table has columns for Name, Image, Organization, and Actions. The 'Name' column is sorted in ascending order. The table lists five execution environments: 'Ansible Engine 2.9 execution environment', 'Control Plane Execution Environment', 'Default execution environment', 'Minimal execution environment', and 'oracle-ai-x-ee'. The 'Image' column shows the container image for each environment. The 'Organization' column shows 'Globally Available' for all environments. The 'Actions' column contains edit and delete icons for each environment.

Name	Image	Organization	Actions
Ansible Engine 2.9 execution environment	registry.redhat.io/ansible-automation-platform-2l/ee-29-rhel8:latest	Globally Available	
Control Plane Execution Environment	registry.redhat.io/ansible-automation-platform-2l/ee-supported-rhel8:latest	Globally Available	
Default execution environment	registry.redhat.io/ansible-automation-platform-2l/ee-supported-rhel8:latest	Globally Available	
Minimal execution environment	registry.redhat.io/ansible-automation-platform-2l/ee-minimal-rhel8:latest	Globally Available	
oracle-ai-x-ee	localhost/oracle_ai_x_ee	Globally Available	

For running the playbook we need to create project and select corresponding execution environment. Here the collection code is placed at /var/lib/awx/projects/oracle directory

The screenshot shows the 'Edit Details' page for a project named 'oracle-ai-x-test-si'. The 'Name' field is 'oracle-ai-x-test-si', the 'Description' field is 'oracle', and the 'Organization' field is 'Default'. The 'Execution Environment' dropdown is set to 'oracle-ai-x-ee' and the 'Source Control Credential Type' dropdown is set to 'Manual'. The 'Project Base Path' field is '/var/lib/awx/projects' and the 'Playbook Directory' dropdown is set to 'oracle'. The 'Save' button is highlighted in blue.

Create credentials, hosts and inventory entities. After that create the project template by selecting the previously created project and other entities

The screenshot shows the 'Edit Details' page for a template named 'oracle-ai-x-template'. The left sidebar contains navigation links: Views, Resources (Templates, Credentials, Projects, Inventories, Hosts), Access (Organizations, Users, Teams), and Administration (Credential Types). The main content area has the following fields:

- Name:** oracle-ai-x-template
- Description:** (empty)
- Job Type:** Run
- Prompt on launch:** ☐
- Inventory:** oracle-ai-x-inv
- Prompt on launch:** ☐
- Project:** oracle-ai-x-test-si
- Execution Environment:** oralice-ai-x-ee
- Playbook:** demo_play_ai-x_oracle.yml
- Credentials:** SSH: aixserver
- Prompt on launch:** ☐
- Labels:** (empty)

Click Launch button to run the playbook

The screenshot shows the 'Details' page for the 'oracle-ai-x-template'. The left sidebar is the same as the previous screenshot. The main content area has tabs: Back to Templates, Details (selected), Access, Notifications, Schedules, Jobs, and Survey. The details are as follows:

Name	oracle-ai-x-template	Job Type	run	Organization	Default
Inventory	oracle-ai-x-inv	Project	oracle-ai-x-test-si	Execution Environment	oralice-ai-x-ee
Playbook	demo_play_ai-x_oracle.yml	Forks	0	Verbosity	0 (Normal)
Timeout	0	Show Changes	Off	Job Slicing	1
Created	3/28/2022, 11:35:42 PM by admin				
Last Modified	4/20/2022, 12:34:59 AM by admin				

Below the table, there are sections for Credentials (SSH: aixserver) and Variables (YAML, JSON). At the bottom, there are buttons for Edit, Launch, and Delete.

View the playbook status from stdout console

The screenshot shows the 'Output' page for a job named 'oracle-ai-x-template'. The left sidebar is the same as the previous screenshots. The main content area has tabs: Back to Jobs, Details, and Output (selected). The output is displayed in a console view with the following content:

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

Appendix

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

On the AIX managed host, check if yum and python are installed. Check directories /usr/bin and /opt/freeware/bin/

```
> which yum

no yum in /usr/bin /etc /usr/sbin /usr/ucb /usr/bin/X11 /sbin
/usr/java8_64/jre/bin /usr/java8_64/bin

> which python

no python in /usr/bin /etc /usr/sbin /usr/ucb /usr/bin/X11 /sbin
/usr/java8_64/jre/bin /usr/java8_64/bin
```

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

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

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

- Install and configure yum, 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/ibm/power_aiX

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



The screenshot shows the 'Installation' section of the Ansible Galaxy collection page for 'ibm.power_aiX'. It displays the command `$ ansible-galaxy collection install ibm.power_aiX` with a copy icon. Below this is a note: 'NOTE: Installing collections with ansible-galaxy is only supported in ansible 2.9+'. A red circle highlights the 'Download tarball' link. The 'Install Version' section shows '1.2.1 released 2 months ago (latest)' with a dropdown arrow. The 'Tags' section shows 'infrastructure', 'ibm', 'power', and 'aix'.

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
```

```
[ansible@p208n149 zips]$ ansible-galaxy collection install ibm.power_aiX
Process install dependency map
Starting collection install process
Installing 'ibm.power_aiX:1.2.1' to '/home/ansible/.ansible/collections/ansible_collections/ibm/power_aiX'
```

The above command will install power_aiX collection at location ~/.ansible/collections

For more information regarding ansible collections refer below link

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

- a) Prepare bootstrap.yml using
“~/ansible/collections/ansible_collections/ibm/power_aix/playbooks/demo_bootstrap.yml”

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

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

# INSTALL / UPDATE Python on inventory host
- name: "Bootstrap Python on AIX"
  hosts: all
  gather_facts: no
  collections:
  - ibm.power_aix
  tasks:

- import_role:
  name: power_aix_bootstrap
  vars:
    pkgtype: "python"
```

- b) Prepare ansible.cfg, inventory files for playbook execution. “p227n241” is AIX managed host mentioned in inventory file. Update the “roles_path” to power-aix collection roles directory.

```
$ cat ansible.cfg
[defaults]
inventory = ./inventory
interpreter_python = /usr/bin/python
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
p227n241
```

c) Executing bootstrap playbook

```
$ ansible-playbook demo_bootstrap.yml
```

d) Bootstrap playbook creates below files in user home directory. You can do a cleanup if needed

#+ cleanup of files created in \$HOME

```
(cd $HOME; rm -f rpm.rte yum_bundle.tar yum_installer.sh
```

Option B, installing them manually onto the managed host.

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

Please go through the README-yum file for instructions

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