Lab: Installing Ansible

Introduction:

Ansible is an **agentless** automation tool that by default manages machines over the **SSH** protocol. Once installed, Ansible does not add a database, and there will be no daemons to start or keep running. You only need to install it on one machine and it can manage an entire fleet of remote machines from that central point. When Ansible manages remote machines, it does not leave software installed or running on them.

In this lab exercise you will learn to Install and Configure the **pre-requisites** for Deploying and Managing Ansible.

Prerequisites:

Installing Ansible on a **Controller Node**, which uses **SSH** (by default) to communicate with your managed nodes (those end devices you want to automate).

Control Node Requirements:

Ansible can be run from any machine with **Python 2** (version 2.7) or **Python 3** (versions 3.5 and higher) installed. This includes Red Hat, Debian, CentOS, macOS, any of the BSDs, and so on. Windows is not supported for the Control Node.

Managed Node Requirements:

On Managed nodes, you need a way to communicate, which is normally **SSH**. By default, this uses SFTP. If that's not available, you can switch to SCP in ansible.cfg. You also need **Python2** (version 2.6 or later) or **Python 3** (version 3.5 or later).

Objectives:

- **Installing Ansible.**
- **Verifying Ansible Installation.**
- **View the Default Ansible Config Files.**
- **Configuring SSH Password-less Access.**
- **Creating Custom Ansible Config File.**

Student Material – Do Not Re-distribute. For any queries contact:
naushad.p.tech@gmail.com or https://www.linkedin.com/in/naushadpasha/

1. Installing Ansible.

1.1 Let's install Ansible Package by executing below command.

```
# dnf -y install ansible
```

Output:

```
[root@eoc-controller ~]#dnf -y install ansible
Extra Packages for Enterprise Linux 8 - x86_64          74 kB/s |  16 MB     03:42
Extra Packages for Enterprise Linux Modular 8 - x86_64  916 kB/s | 733 kB     00:00
Last metadata expiration check: 0:00:01 ago on Sun 28 Jan 2024 01:42:25 AM EST.
Dependencies resolved.
================================================================================
 Package                       Arch        Version              Repository      Size
================================================================================
Installing:
 ansible                       noarch      8.3.0-1.el8          epel            41 M
Installing dependencies:
 ansible-core                  x86_64      2.16.2-1.el8         appstream      4.1 M
 git-core                      x86_64      2.43.0-1.el8         appstream       11 M
 mpdecimal                     x86_64      2.5.1-3.el8          appstream       93 k
 python3.11                    x86_64      3.11.5-2.el8         appstream       30 k
 python3.11-cffi               x86_64      1.15.1-1.el8         appstream      305 k
 python3.11-cryptography       x86_64      37.0.2-5.el8         appstream      1.2 M
 python3.11-libs               x86_64      3.11.5-2.el8         appstream       11 M
 python3.11-pip-wheel          noarch      22.3.1-4.el8         appstream      1.4 M
 python3.11-ply                noarch      3.11-1.el8           appstream      138 k
 python3.11-pycparser          noarch      2.20-1.el8           appstream      157 k
 python3.11-pyyaml             x86_64      6.0-1.el8            appstream      228 k
 python3.11-setuptools-wheel   noarch      65.5.1-2.el8         appstream      720 k
 sshpass                       x86_64      1.09-4.el8           appstream       30 k
Installing weak dependencies:
 python3-jmespath              noarch      0.9.0-11.el8         appstream       45 k

Transaction Summary
================================================================================
Install  15 Packages
```

2. Verifying Ansible Installation

2.1 Let's verify that Ansible is installed with its version details.

```
# ansible --version
```

Output:

```
[root@eoc-controller ~]#ansible --version
ansible [core 2.16.2]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/root/.ansible/plugins/modules', '/usr/share/ansible
/plugins/modules']
  ansible python module location = /usr/lib/python3.11/site-packages/ansible
  ansible collection location = /root/.ansible/collections:/usr/share/ansible/collection
s
  executable location = /usr/bin/ansible
  python version = 3.11.5 (main, Jan 12 2024, 23:13:15) [GCC 8.5.0 20210514 (Red Hat 8.5
.0-21)] (/usr/bin/python3.11)
  jinja version = 3.1.2
  libyaml = True
```

2.2 Let's verify the installation of ansible-python on the localhost by using the setup mode.

```
# ansible localhost -m setup | grep ansible_python_version
```

Output:

```
[root@eoc-controller ~]#ansible localhost -m setup | grep ansible_python_version
        "ansible_python_version": "3.11.5",
```

Student Material – Do Not Re-distribute. For any queries contact:
naushad.p.tech@gmail.com or https://www.linkedin.com/in/naushadpasha/

3. Check the default Ansible Config Files

3.1 Let's view the default configuration

```
# ansible-config view
```

Output:

```
[root@eoc-controller ~]# ansible-config view
# Since Ansible 2.12 (core):
# To generate an example config file (a "disabled" one with all default settings, commented out):
#               $ ansible-config init --disabled > ansible.cfg
#
# Also you can now have a more complete file by including existing plugins:
# ansible-config init --disabled -t all > ansible.cfg

# For previous versions of Ansible you can check for examples in the 'stable' branches of each versi
# Note that this file was always incomplete  and lagging changes to configuration settings

# for example, for 2.9: https://github.com/ansible/ansible/blob/stable-2.9/examples/ansible.cfg
```

3.2 Let's view the content of the default configuration file /etc/ansible/ansible.cfg.

```
# cat /etc/ansible/ansible.cfg
```

Output:

```
[root@eoc-controller ~]# cat /etc/ansible/ansible.cfg
# Since Ansible 2.12 (core):
# To generate an example config file (a "disabled" one with all default settings, commented out):
#               $ ansible-config init --disabled > ansible.cfg
#
# Also you can now have a more complete file by including existing plugins:
# ansible-config init --disabled -t all > ansible.cfg

# For previous versions of Ansible you can check for examples in the 'stable' branches of each versio
n
# Note that this file was always incomplete  and lagging changes to configuration settings

# for example, for 2.9: https://github.com/ansible/ansible/blob/stable-2.9/examples/ansible.cfg
```

```
# ansible-config init view
```

```
[root@eoc-controller ~]#ansible-config init view
[defaults]
# (boolean) By default Ansible will issue a warning when received from a task action (module or
# These warnings can be silenced by adjusting this setting to False.
action_warnings=True

# (list) Accept list of cowsay templates that are 'safe' to use, set to empty list if you want
cowsay_enabled_stencils=bud-frogs, bunny, cheese, daemon, default, dragon, elephant-in-snake, e

# (string) Specify a custom cowsay path or swap in your cowsay implementation of choice
cowpath=

# (string) This allows you to chose a specific cowsay stencil for the banners or use 'random' t
cow_selection=default

# (boolean) This option forces color mode even when running without a TTY or the "nocolor" sett
force_color=False

# (path) The default root path for Ansible config files on the controller.
home=~/.ansible

# (boolean) This setting allows suppressing colorizing output, which is used to give a better i
nocolor=False

# (boolean) If you have cowsay installed but want to avoid the 'cows' (why????), use this.
nocows=False
```

Note: Press **q** to exit out.

Student Material – Do Not Re-distribute. For any queries contact:
naushad.p.tech@gmail.com or https://www.linkedin.com/in/naushadpasha/

3.3 Let's view the content of the default inventory file /etc/ansible/hosts.

```
# less /etc/ansible/hosts
```

Output:

```
[root@eoc-controller ~]# less /etc/ansible/hosts
# This is the default ansible 'hosts' file.
#
# It should live in /etc/ansible/hosts
#
#   - Comments begin with the '#' character
#   - Blank lines are ignored
#   - Groups of hosts are delimited by [header] elements
#   - You can enter hostnames or ip addresses
#   - A hostname/ip can be a member of multiple groups

# Ex 1: Ungrouped hosts, specify before any group headers:

## green.example.com
## blue.example.com
## 192.168.100.1
## 192.168.100.10

# Ex 2: A collection of hosts belonging to the 'webservers' group:

## [webservers]
## alpha.example.org
## beta.example.org
## 192.168.1.100
```

Truncated…

Note: Press **q** to come out from the prompt

Note: This is the default location of the inventory file, script, or directory that Ansible will use to determine what hosts it has available to talk

3.4 Let's list the managed hosts in the **/etc/ansible/hosts** inventory file

```
# ansible all --list-hosts
```

Output:

```
[root@eoc-controller ~]# ansible all --list-hosts
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost
does not match 'all'
  hosts (0):
```

Note: We will be working on building the inventory file later in the other exercises.

4. Configuring SSH Password-less Access

4.1 Let's switch as admin user.

```
# su - admin
```

Output:

```
[root@eoc-controller ~]# su - admin
[admin@eoc-controller ~]$
```

Student Material – Do Not Re-distribute. For any queries contact:
naushad.p.tech@gmail.com or https://www.linkedin.com/in/naushadpasha/

4.2 Let's generate SSH key-pair for admin user.

```
# ssh-keygen -t rsa -N ''
```

Output:

```
[admin@eoc-controller ~]$ ssh-keygen -t rsa -N ''
Generating public/private rsa key pair.
Enter file in which to save the key (/home/admin/.ssh/id_rsa):
Created directory '/home/admin/.ssh'.
Your identification has been saved in /home/admin/.ssh/id_rsa
Your public key has been saved in /home/admin/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:ZHTc+ZOqQ0LjoGnOZzs9GIJv7jQmWMAmonCKISqcM7I admin@eoc-controller
The key's randomart image is:
+---[RSA 3072]----+
|        .... .   |
|.      . .. o    |
|Bo.     o    . . |
|@=.   .oo     +  |
|B=.. o +S.   . . |
|o++ = . o . .    |
|E .=+. + o .     |
|   +=.= o o      |
|   +oo.o . .     |
+----[SHA256]-----+
```

4.3 Let's gather ssh public keys, by executing below command.

```
# ssh-keyscan eoc-controller eoc-node1 eoc-node2 eoc-\
node3 >> ~/.ssh/known_hosts
```

Output:

```
[admin@eoc-controller ~]$ssh-keyscan eoc-controller eoc-node1 eoc-node2 eoc-node3 >> ~/.ssh/
known_hosts
# eoc-node2:22 SSH-2.0-OpenSSH_8.0
# eoc-controller:22 SSH-2.0-OpenSSH_8.0
# eoc-controller:22 SSH-2.0-OpenSSH_8.0
# eoc-controller:22 SSH-2.0-OpenSSH_8.0
# eoc-node1:22 SSH-2.0-OpenSSH_8.0
# eoc-node1:22 SSH-2.0-OpenSSH_8.0
# eoc-node2:22 SSH-2.0-OpenSSH_8.0
# eoc-node2:22 SSH-2.0-OpenSSH_8.0
# eoc-node1:22 SSH-2.0-OpenSSH_8.0
# eoc-node3:22 SSH-2.0-OpenSSH_8.0
# eoc-node3:22 SSH-2.0-OpenSSH_8.0
# eoc-node3:22 SSH-2.0-OpenSSH_8.0
```

4.4 Let's copy the keys to all of the nodes, by executing below command.

Note: Password linux when prompted.

```
# ssh-copy-id eoc-controller
```

Output:

```
[admin@eoc-controller ~]$ ssh-copy-id eoc-controller
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/admin/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are alread
y 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
admin@eoc-controller's password:

Number of key(s) added: 1

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

Student Material – Do Not Re-distribute. For any queries contact:
naushad.p.tech@gmail.com or https://www.linkedin.com/in/naushadpasha/

```
# ssh-copy-id eoc-node1
```

Output:

```
[admin@eoc-controller ~]$ ssh-copy-id eoc-node1
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/admin/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are alread
y 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
admin@eoc-node1's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'eoc-node1'"
and check to make sure that only the key(s) you wanted were added.
```

```
# ssh-copy-id eoc-node2
```

Output:

```
[admin@eoc-controller ~]$ ssh-copy-id eoc-node2
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/admin/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are alread
y 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
admin@eoc-node2's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'eoc-node2'"
and check to make sure that only the key(s) you wanted were added.
```

Output:

```
# ssh-copy-id eoc-node3
```

```
[admin@eoc-controller ~]$ ssh-copy-id eoc-node3
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/admin/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are alread
y 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
admin@eoc-node3's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'eoc-node3'"
and check to make sure that only the key(s) you wanted were added.
```

5. Creating Custom Ansible Config File

5.1 In the home directory of the user create a **ansible.cfg** file with below entries.

```
# cat > ~/.ansible.cfg <<EOF
[defaults]
inventory = ~/org-infra
roles_path = ~/roles
remote_user = admin
become = true
EOF
```

Student Material – Do Not Re-distribute. For any queries contact:
naushad.p.tech@gmail.com or https://www.linkedin.com/in/naushadpasha/

inventory = ~/org-infra ---> path of hostfile.
roles_path = ~/roles ---> additional paths to search for roles in, colon separated.
remote_user = admin ---> This is the default username ansible will connect as for /usr/bin/ansible-playbook. Note that /usr/bin/ansible will always default to the current user if this is not defined.
become=true---> The equivalent of adding sudo: or su: to a play or task, set to true/yes to activate privilege escalation. The default behavior is no.

5.2 Let's edit the **/home/admin/.ansible.cfg** file in a text editor.

```
# cat >> ~/.ansible.cfg <<EOF
[privilege_escalation]
become=True
become_method=sudo
become_user=root
become_ask_pass=False
EOF
```

become=True---> The equivalent of adding sudo: or su: to a play or task, set to true/yes to activate privilege escalation. The default behavior is no.
become_method=sudo---> Set the privilege escalation method. The default is sudo, other options are su, pbrun, pfexec.
become_user=sudo ---> The equivalent to ansible_sudo_user or ansible_su_user, allows to set the user you become through privilege escalation. The default is 'root'.
become_ask_pass=False---> Ask for privilege escalation password, the default is False.

5.3 Verify the contents of the file ~/.ansible.cfg.

```
# cat ~/.ansible.cfg
```

Output:

```
[admin@eoc-controller ~]$ cat ~/.ansible.cfg
[defaults]
inventory = ~/org-infra
roles_path = ~/roles
remote_user = admin
become = true
[privilege_escalation]
become=True
become_method=sudo
become_user=root
become_ask_pass=False
```

Student Material – Do Not Re-distribute. For any queries contact:
naushad.p.tech@gmail.com or https://www.linkedin.com/in/naushadpasha/

5.4 Verify that the ansible is using **/home/admin/ansible.cfg** as the configuration file.

```
# ansible --version
```

Output:

```
[admin@eoc-controller ~]$ansible --version
ansible [core 2.16.2]
  config file = /home/admin/.ansible.cfg
  configured module search path = ['/home/admin/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3.11/site-packages/ansible
  ansible collection location = /home/admin/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.11.5 (main, Jan 12 2024, 23:13:15) [GCC 8.5.0 20210514 (Red Hat 8.5.0-21)] (/usr/bin/python3.11)
  jinja version = 3.1.2
  libyaml = True
```

Student Material – Do Not Re-distribute. For any queries contact:
naushad.p.tech@gmail.com or https://www.linkedin.com/in/naushadpasha/