Ansible Control Node, Inventory & Managed Nodes

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

RHCE (EX294) Session - 3: (Topics Covered)

1. Installation of Ansible control node (Ansible Server).
2. Inventory file (Static Inventory).
3. Connection with the managed nodes.

---

1. Installation of Ansible Control Node (Ansible Server):

 # hostnamectl set-hostname ansible-server.nehraclasses.local
 # vim /etc/hosts
192.168.1.14 ansible-server.nehraclasses.local localhost
192.168.1.16 node1.nehraclasses.local node1
192.168.1.17 node2.nehraclasses.local node2
 # ping node1
 # ping node2
 # dnf install -y ansible-core

---

2. Inventory File (Static Inventory):

 # ansible --version
 # ls -lh /etc/ansible/
ansible.cfg = configuration file
hosts = default inventory file of ansible

Inventory File:

The Ansible inventory file defines the hosts and groups of hosts upon which commands, modules, and tasks in a playbook operate. The file can be in one or many formats depending on your Ansible environment and plugins. Common formats include INI and YAML.

Types of Inventories in Ansible:

1. Static Inventory
2. Dynamic Inventory

1. Static Inventory:

In Ansible, a static inventory file is a plain text file that contains a list of managed hosts declared under a host group using either hostnames or IP addresses.

Inventory File Characteristics/Features:

(a) Name: Any name whatever you want
(b) Path: Not fixed path required
(c) Root Access: Not necessary
(d) Language/Format: INI (initialization file), YAML

Ansible Control Node, Inventory & Managed Nodes

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

Custom Ansible Inventory File: Rules for writing inventory files:
 # su - vikasnehra
$ mkdir inventory
$ cd inventory

$ vim nodes
 # we can create groups using names in square brackets.
[web]
192.168.1.16
192.168.1.17

 # you can mention hostnames as well if DNS is available in your environment.
[app]
app1.nehraclasses.local
app2.nehraclasses.local
app3.nehraclasses.local
[dev]
192.168.1.10
192.168.1.11

 # if we have large number of hosts or series of IP addresses or complete subnet.
[database]
192.168.10.[1:20]

 # if we have large number of hosts having hostname in a sequence.
[sandbox]
server[1:9].nehraclasse.local

 # we can also create group using groups as well. E.g if you want to use web & dev groups either you can mention both in playbooks or create group like below.
[test:children]
web
dev

Save & close the file. Then check the file for errors, whether it will work or not?
 # ansible web --list-hosts

It will not show you the host machine names because ansible is currently not able to read the nodes inventory file created by you. You have to use option -i with the command and specify the location of the inventory file.
$ ansible web --list-hosts -i ~/inventory/nodes

$ ansible web --list-hosts -i ~/inventory
$ ansible dev --list-hosts -i ~/inventory
$ ansible web:dev --list-hosts -i ~/inventory
$ ansible database --list-hosts -i ~/inventory
$ ansible test --list-hosts -i ~/inventory
$ ansible all --list-hosts -i ~/inventory

We can create multiple inventory files in the same environment.
$ vim os
[rhel]
192.168.2.1
192.168.2.2
192.168.2.3

[ubuntu]
vm1
vm2
vm3

Save and close the file.

Now check the host details from this os inventory file.
$ ansible ubuntu --list-hosts -i ~/inventory
$ ansible all --list-hosts -i ~/inventory

In case of multiple inventory files, ansible will read these files in an alphabetical order one after other. Other users working on control node can also create their own inventory files, if needed.

To use default inventory file, make an entry in */etc/ansible/hosts* file for the nodes machines. (sudo or root access required)
$ su -
 # vim /etc/ansible/hosts
[nehraclasses]
192.168.20.2
192.168.20.8

Now, list the node machines.
 # ansible nehraclasses --list-hosts
 # ansible all --list-hosts

Ansible Control Node, Inventory & Managed Nodes

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

It will only show you the machines present in default inventory file. In order to list the machines available in the inventory file of 'vikasnehra' user we have to specify the file location using option -i in the command.
 # ansible all --list-hosts -i /home/vikasnehra/inventory

 # ll /etc/ansible/
ansible.cfg = configuration file
hosts = default inventory file of ansible

To make your custom inventory file as default inventory file, define the same in ansible.cfg file located in /etc/ansible/ directory.
 # ansible --version
 # vim /etc/ansible/ansible.cfg
[defaults]
inventory = /home/vikasnehra/inventory

Now, the default inventory has been changed.
 # ansible web --list-hosts

To list all the machines in all the groups.
 # ansible all --list-hosts

Let's change it back to default one by commenting lines added previously.
 # vim /etc/ansible/ansible.cfg
#[defaults]
#inventory = /home/vikasnehra/inventory

Verify the changes.
 # ansible all --list-hosts

Or you can create your own custom ansible configuration file where you can define this option as well as other options.
 # su - vikasnehra
$ ansible --version
$ vim ansible.cfg
[defaults]
inventory = /home/vikasnehra/inventory/nodes

Verify the changes.
$ ansible --version

Ansible Control Node, Inventory & Managed Nodes

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

If the ansible configuration file is present in any other location, in that case we have to set the ANSIBLE_CONFIG variable for the same.
$ cp ansible.cfg /tmp/ansible-vikasnehra.cfg
$ export ANSIBLE_CONFIG=/tmp/ansible-vikasnehra.cfg
$ echo $ANSIBLE_CONFIG

Verify the same.
$ ansible --version

You can put this variable in profile files for permanent changes.
$ vim .bashrc
export ANSIBLE_CONFIG=/tmp/ansible-vikasnehra.cfg

By default, ansible looks for the locations of the ansible configuration files in the following order:
(a) exported value
(b) present working directory
(c) user home directory
(d) /etc/ansible/ansible.cfg

Now, lets remove this variable.
$ unset *ANSIBLE_CONFIG*
$ vim .bashrc
#export ANSIBLE_CONFIG=/tmp/ansible-vikasnehra.cfg

---

3. Connection With The Managed Nodes:
SSH connection is required in order to make a connection with the managed nodes.
Types of SSH Authentication for connections:
(a) Password Based
(b) Password-less or Key Based
(c) Inventory Based

(a) Password Based:
$ su - vikasnehra

Try to ping other nodes using ansible from this account where we don't have password-less authentication.
$ ansible all -m ping
$ ansible web -m ping

You have to use -k option with the command for SSH password prompt.

$ ansible web -m ping -k

It will throw an error because of ssh key checking option. You can disable it in ansible configuration file.

$ vim ansible.cfg
[defaults]
inventory = /home/vikasnehra/inventory/
host_key_checking = false

Now, you can use user password for ansible connection.

$ ansible web -m ping -k

If you want to use some other user account (other than from which you are running this command) using his account password.

$ ansible web -m ping -k -u demouser
$ ansible all -m ping -k -u demouser

(b) Passwordless or Key Based:

Create a local user account on all the nodes and add this user to sudoers file as well.

 # useradd nehraclasses
 # echo "redhat" | passwd --stdin nehraclasses
 # echo "nehraclasses ALL=(ALL) NOPASSWD:ALL" | sudo tee /etc/sudoers.d/nehraclasses
 # su - nehraclasses

Generate SSH public and private keys on the ansible server and copy the public key file to other nodes.

$ ssh-keygen
$ ssh-copy-id nehraclasses@node1
$ ssh nehraclasses@node1
$ ssh-copy-id nehraclasses@node2
$ ssh nehraclasses@node2

On Ansible control node and create ansible.cfg file with beneath content.

$ mkdir ~/automation && cd ~/automation/
$ vim ansible.cfg
[defaults]
inventory = ./inventory
host_key_checking = false
remote_user = nehraclasses
ask_pass = False

[privilege_escalation]
become=true
become_method=sudo
become_user=root
become_ask_pass=False

$ vim ~/automation/inventory
[web]
192.168.1.16
192.168.1.17

Now, check whether you are able to ping your nodes machines using ansible or not.
$ cd ~/automation/
$ ansible all -m ping

(c) Inventory Based:
Useful in case same if all machines are having different user accounts
 # useradd nehra
 # echo "redhat" | passwd --stdin nehra
 # su - nehra
$ mkdir ~/automation && cd ~/automation/

$ vim ansible.cfg
[defaults]
inventory = ./inventory
host_key_checking = false

Define the authentication information in the inventory file.
$ vim inventory
[web]
192.168.1.16 ansible_ssh_user=vikasnehra ansible_ssh_pass=redhat
192.168.1.17 ansible_ssh_user=nehraclasses ansible_ssh_pass=redhat
[dev]
192.168.1.10

Now, check whether you are able to ping your nodes machines using ansible or not.
$ ansible all -m ping -i ~/automation/inventory

We can use inventory variables as well for this purpose.
$ vim inventory

[web]
192.168.1.16
192.168.1.17

[dev]
192.168.1.10

 # define password for the entire group of machine if user account is present on all machines.
[web:vars]
ansible_ssh_user=vikasnehra
ansible_ssh_pass=redhat

Now, check whether you are able to ping your nodes machines using ansible or not.
$ ansible all -m ping -i ~/automation/inventory

Its not a good practice to keep passwords in simple text files. In the upcoming sessions, we will learn how can we encrypt these passwords stored in plain text using ansible vault.

Thank You