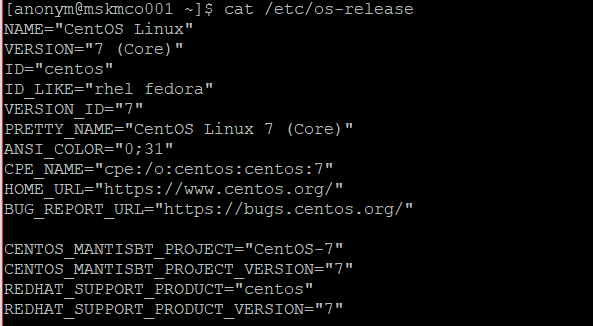
My Environment setup:

* Controlling Machine – Ansible Server ( master ) :

OS:



IP Address: 192.168.65.136 Hostname: mskmco001 User: anonym

* Remote Nodes ( slaves ) :

Slave 1:

IP Address: 192.168.65.132 Hostname: sskmco001 User: anonym

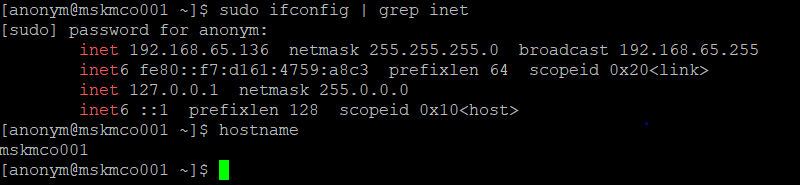
Slave 2:

IP Address: 192.168.65.133 Hostname: sskmco002 User:anonym

Step 1: Installing Controlling Machine – Ansible

1. Before installing ‘Ansible’ on the server, let’s first verify the details of the server like hostname and IP Address. Login into server as a root user and execute the below command to confirm system settings that we’re going to use for this setup.

# sudo ifconfig | grep inet



2. Once you confirm your system settings, it’s time to install ‘Ansible’ on the system.

To get Ansible for CentOS 7, first ensure that the CentOS 7 EPEL repository is installed:

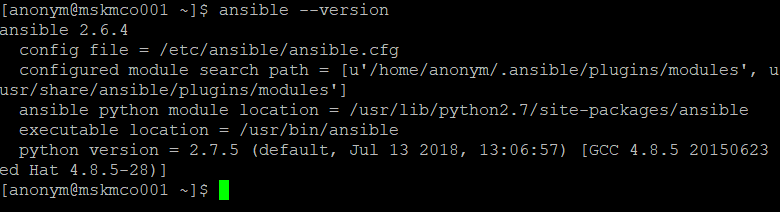
# sudo yum install epel-release

After configuring epel repository, you can install Ansible using the following command.

# sudo yum install ansible -y

After installation, verify the version by executing below command.

# ansible --version



Step 2: Preparing SSH Keys to Remote Hosts

3. To perform any deployment or management from the localhost to remote host first we need to create and copy the ssh keys to the remote host. In every remote host there will be a user account anonym (in your case may be different user).

First let we create a SSH key using below command and copy the key to remote hosts.

# ssh-keygen -t rsa -b 4096 -C "anonym@mskmco001.com"

A public/private key pair gets generated.

Note: Do not enter any passphrase while generating the key

4. After creating SSH Key successfully, now copy the created key to all three remote servers.

# ssh-copy-id anonym@sskmco001

# ssh-copy-id anonym@sskmco002

Note: you can use ip address instead of hostname too in the above command.

5. After copying all SSH Keys to remote host, now perform a ssh key authentication on all remote hosts to check whether authentication working or not.

# ssh anonym@sskmco001

# ssh anonym@sskmco002

You should be able to login to the slaves machine without entering any password via ssh.

Step 3: Creating Inventory File for Remote Hosts

Inventory file, this file holds the host information like which host we need to get connect from local (master) to remote(slaves).

Default inventory file will be under /etc/ansible/hosts.

6. Now let’s add these two hosts to inventory file. Open and edit file using your favourite editor, Here I use vim.

# sudo vim /etc/ansible/hosts

Add the following two hosts hostname/IP address.

[slave-machines]

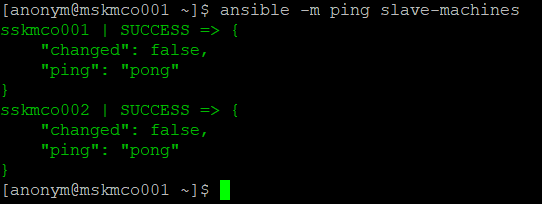
sskmco001

sskmco002

Note: The ‘slave-machines’ in the bracket are called groups. Hosts can be in multiple groups and groups can configure parameters for all of their members.

7. Now it is time to check all our servers by just doing a ping from our master. To perform the action we need to use the command ‘ansible’ with options ‘-m’ (module) and ‘-all’ (group of servers).

# ansible -m ping slave-machines OR # ansible -m ping all



8. Now, we are using a module called ‘command‘, which is used to execute list of commands (like, df, free, uptim, etc.) on all selected remote hosts at one go, for example watch out few examples shown below.

a. To check the partitions on all remote hosts

# ansible -m command -a "df -h" slave-machines

b. Check memory usage on all remote hosts.

# ansible -m command -a "free -mt" slave-machines

c. Check Uptime for all 3 servers.

# ansible -m command -a "uptime" slave-machines

**d**. Check Architecture and Hostname.

# ansible -m command -a "arch" slave-machines

# ansible -m shell -a "hostname" slave-machines

**e**. If we need to dump the output to any file we can redirect as below.

# ansible -m command -a "df -h" slave-machines > /tmp/df\_output.txt

