**Installing Ansible Server**

There is no official Ansible repository for RPB based clones, but we can install Ansible by enabling epel repository using RHEL/CentOS 6. X, 7. X using the currently supported fedora distributions.

# rpm -Uvh http://download.fedoraproject.org/pub/epel/6/i386/epel-release-6-8.noarch.rpm

**Output:**

Retrieving http://download.fedoraproject.org/pub/epel/6/i386/epel-release-6-8.no             arch.rpm

warning: /var/tmp/rpm-tmp.nHoRHj: Header V3 RSA/SHA256 Signature, key ID 0608b89             5: NOKEY

Preparing...                ########################################### [100%]

package epel-release-6-8.noarch is installed

After configuring epel repository, you can now install Ansible using yum with the below command.

# sudo yum install ansible -y

**Output:**

Loaded plugins: fastestmirror, security

Setting up Install Process

Determining fastest mirrors

epel/metalink                                            | 4.3 kB     00:00

 \* base: centosmirror.go4hosting.in

 \* epel: epel.mirror.net.in

 \* extras: centosmirror.go4hosting.in

 \* updates: centosmirror.go4hosting.in

Resolving Dependencies

.

.

.

Installed:

  ansible.noarch 0:1.9.4-1.el6

Dependency Installed:

  PyYAML.x86\_64 0:3.10-3.1.el6                   libyaml.x86\_64 0:0.1.3-4.el6\_6

  python-babel.noarch 0:0.9.4-5.1.el6            python-crypto2.6.x86\_64 0:2.6.1-2.el6

  python-httplib2.noarch 0:0.7.7-1.el6           python-jinja2.x86\_64 0:2.2.1-2.el6\_5

  python-keyczar.noarch 0:0.71c-1.el6            python-pyasn1.noarch 0:0.0.12a-1.el6

  python-simplejson.x86\_64 0:2.0.9-3.1.el6       sshpass.x86\_64 0:1.05-1.el6

Complete!

**Verifying the Installation**

After configuring epel repository, you can now install Ansible using yum with the below command.

# ansible --version

ansible 1.9.4

  configured module search path = None

**Preparing SSH Keys to Remote Hosts**

To perform any deployment or up-gradation from the ansible server, for every host, there should be a user account to communicate. Also, we need to copy the ssh keys from the Anisble server to the remote host for password-less connection.

First, let us create an SSH key using the below command and copy the key to remote hosts.

# ssh-keygen -t rsa -b 4096 -C "ansible.hanuman.com"

**Generating Public/Private rsa Key Pair**

Enter file in which to save the key (/root/.ssh/id\_rsa):

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in ansible\_key.

Your public key has been saved in ansible\_key.pub.

The key fingerprint is:

28:ae:0c:8d:91:0a:fa:ac:2f:e2:8c:e5:fd:28:4b:c6 ansible.hanuman.com

The key's randomart image is:

+--[ RSA 4096]----+

|                 |

|                 |

|                 |

| .     .         |

|+   . . S        |

|+= . .           |

|= E .            |

|=X.o .           |

|=\*Ooo..          |

+-----------------+

After creating SSH Key success, now copy the created key to all the two remote servers, We need a user to do ansible here for a demo and I am using root user from where we can perform the ansible tasks.

# ssh-copy-id root@192.168.87.156

**Output:**

root@192.168.87.156's password:

Now try logging into the machine, with "ssh 'root@192.168.87.156'", and check in:

  .ssh/authorized\_keys

to make sure we haven't added extra keys that you weren't expecting.

# ssh-copy-id root@192.168.87.157

**Output:**

root@192.168.87.157's password:

Now try logging into the machine, with "ssh 'root@192.168.87.157'", and check in:

  .ssh/authorized\_keys

to make sure we haven't added extra keys that you weren't expecting.

**Copy SSH Key Second Remote Host**

After copying all SSH Keys to remote host, now perform an ssh key authentication on all remote hosts to check whether authentication working or not run below commands to test.

# ssh root@192.168.87.156

[ansible@localhost ~]#

Connection to 192.168.87.156 closed.

# ssh root@192.168.87.157

[ansible@localhost ~]#

**Creating Inventory File for Remote Hosts**

Inventory file, This file has information about the hosts for which host we need to get connected from local to remote. The default configuration file will be under /etc/ansible/hosts.

Now, we will add the two  nodes to configuration file. Open and edit file using your favorite editor, Here we are using vim.

# sudo vim /etc/ansible/hosts

Add the following two hosts IP address..

[webservers]

192.168.87.156

192.168.87.157

**Note:**  [webservers] in the brackets indicates as group names, it is used to classify the nodes and group them and  to controlling at what times and for what reason.

**To Test if Ansible is Working or Not**

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

# ansible -m ping webservers

**Output:**

[root@localhost ~]# ansible -m ping webservers

192.168.87.157 | success >> {

"changed": false,

"ping": "pong"

}

192.168.87.156 | success >> {

"changed": false,

"ping": "pong"

}

OR

# ansible -m ping -all

**Output:**

[root@localhost ~]# ansible -m ping webservers

192.168.87.157 | success >> {

    "changed": false,

    "ping": "pong"

}

192.168.87.156 | success >> {

    "changed": false,

    "ping": "pong"

}

Now, here we are using another module called ‘command’, which is used to execute a list of shell commands (like, df, free, uptime, etc.) on all selected remote hosts at one go. For demo you can execute the below commands.

**Check the Partitions on all Remote Hosts**

# ansible -m command -a "df -h" webservers

**Output:**

192.168.87.156 | success | rc=0 >>

 Filesystem            Size  Used Avail Use% Mounted on

 /dev/mapper/VolGroup-lv\_root

 18G  2.0G   15G  12% /

 tmpfs                 491M     0  491M   0% /dev/shm

 /dev/sda1             477M   42M  411M  10% /boot

192.168.87.157 | success | rc=0 >>

 Filesystem            Size  Used Avail Use% Mounted on

 /dev/mapper/VolGroup-lv\_root

 18G  2.0G   15G  12% /

 tmpfs                 491M     0  491M   0% /dev/shm

 /dev/sda1             477M   42M  411M  10% /boot

**Check Memory Usage for all Webservers**

# ansible -m command -a "free -mt" webservers

**Output:**

192.168.87.156 | success | rc=0 >>

 total       used       free     shared    buffers     cached

 Mem:           981        528        453          0         39        322

 -/+ buffers/cache:        166        815

 Swap:         2047          0       2047

 Total:        3029        528       2501

192.168.87.157 | success | rc=0 >>

 total       used       free     shared    buffers     cached

 Mem:           981        526        455          0         39        322

 -/+ buffers/cache:        164        817

 Swap:         2047          0       2047

 Total:        3029        526       2503

**Redirecting the Output to a File**

# ansible -m shell -a "service httpd status" webservers > service\_status.txt

**Output:**

# cat service\_status.txt

 192.168.87.156 | FAILED | rc=3 >>

 httpd is stopped

192.168.87.157 | FAILED | rc=3 >>

 httpd is stopped

**To Shut down the Remote Servers**

#ansible -m shell -a "init 0" webservers

**OutPut:**

192.168.87.157 | success | rc=0 >>

192.168.87.156 | success | rc=0 >>