**Ansible installation in Linux Master**

* You can consider **ansible ad-hoc commands as shell commands** and a **playbook as a shell script.**

**Steps to install Ansible in Master node:**

1. **Sudo su**
2. Create hostname **“hostname Ansible-Master”**
3. Change hostname permanently **“ vi /etc/hostname”**
4. To use hostname follow bellow commands

[root@ip-172-31-19-198 ec2-user]# **exit**

exit

[ec2-user@ip-172-31-19-198 ~]$ **sudo su**

[root@Ansible-Master ec2-user]#

1. Now create user **“useradd ansimaster**”
2. For to see howmany users are there use **“cat /etc/passwd”**
3. To create password for user use **“passwd ansimaster”**

[root@Ansible-Master ec2-user]# **passwd ansimaster**

**Changing password for user ansimaser.**

New password: **pavan@140697**

Retype new password: **pavan@140697**

passwd: **all authentication tokens updated successfully.**

1. I need to give ROOT permissions to USER

**Visudo**

Add user

root ALL=(ALL) ALL

ansimaster ALL=(ALL) NOPASSWD: ALL

1. Now provide Password Authentication to user

**vi /etc/ssh/sshd\_config**

# To disable tunneled clear text passwords, change to no here!

PasswordAuthentication yes

#PermitEmptyPasswords no

#PasswordAuthentication no

1. Now restart the sshd service

**service sshd restart/reload**

1. Now change as USER

**Sudo su – ansimaster**

[ec2-user@Ansible-Master ~]$ **sudo su - ansimaster**

[**ansi@**Ansible-Master ~]$

1. After changed to USER generate SSH keys

**ssh-keygen** and then click on 3 times on **enter**

**To Install Ansible we need Pre-requisites:**

1. Python and pip

Python already there and to install pip “**sudo yum install python-pip**”

1. Now install ANSIBLE

(sudo amazon-linux-extras install ansible2)

**RedHat/CentOS:** sudo yum install ansible

**Fedora:** sudo dnf install ansible

**Ubuntu:** sudo apt-get install ansible

**PIP:** sudo pip install ansible

Sudo yum install epel-release

Sudo yum install python-pip

sudo pip install ansible==2.4 or

sudo pip install --upgrade ansible

1. I am used **“sudo pip install ansible”**
2. Check version **“ansible –version”**

**Note:** if you used yum command by default it creates directory **“/etc/ansible”** but we used pip so I need to create MANUALLY.

1. **Sudo mkdir /etc/ansible**
2. **cd /etc/ansible**
3. I need **to create configuration file** in this directory **“sudo touch ansible.cfg”**

Go to this link <https://docs.ansible.com/ansible/latest/installation_guide/intro_configuration.html>

or

click on <https://github.com/ansible/ansible/blob/devel/examples/ansible.cfg>

and copy the content paste in **“sudo vi** **ansible.cfg”**

1. Create INVENTORY/HOST file **“sudo touch hosts”**

**Note:**

In host file we provide **IP address** and create **Groups.**

**Create Nodes/Slaves**

I created 2 instances. One is “Linux” and “Ubuntu”.

Create user in each instance.

1. **Sudo su**
2. Create hostname **“hostname Ansible-Manage-node01/02”**
3. Change hostname permanently **“ vi /etc/hostname”**
4. To use hostname follow bellow commands

[root@ip-172-31-19-198 ec2-user]# **exit**

exit

[ec2-user@ip-172-31-19-198 ~]$ **sudo su**

[root@Ansible-Master ec2-user]#

1. Now create user **“useradd ansi-manage01**”

For Ubuntu

**Useradd -m -d /home/ansi-manage02 ansi-manage02**

1. For to see howmany users are their use **“cat /etc/passwd”**
2. To create password for user use **“passwd ansi-manage01/02”**

[root@Ansible-Master ec2-user]# **passwd** **ansi-manage01/02**

**Changing password for user ansi-manage01. ===> adapala303**

New password: **Adapal@303 ===> this is for ansi-manage02**

Retype new password: **Adapal@303**

passwd: **all authentication tokens updated successfully.**

1. I need to give ROOT permissions to USER

**Visudo**

Add user

root ALL=(ALL) ALL

ansi-manage01/02 ALL=(ALL) NOPASSWD: ALL

1. Now provide Password Authentication to user

**vi /etc/ssh/sshd\_config**

# To disable tunneled clear text passwords, change to no here!

PasswordAuthentication yes

#PermitEmptyPasswords no

#PasswordAuthentication no

1. Now restart the sshd service

**service sshd restart/reload**

1. Now change as USER

**Sudo su – ansi-manage01/02**

**Now do some tasks in Nodes/Slaves by using Ansible Master**

* Go to “ansible master” instance switch to user -> go to “ cd /etc/ansible” -> sudo vi hosts -> create groups and add “private IP of Instances”.
* From my master I used this command
* In hosts file we need to enter

10.10.11.127 ansible\_ssh\_pass= **adapala303** ansible\_ssh\_user= **ansi-manage01**

* ssh-copy-id ansi-manage01@10.10.11.127

Slave user@Private IP

enter password of slave user “**adapala303”**

* **ssh** [**ansi-manage01@10.10.11.127**](mailto:ansi-manage01@10.10.11.127) **===> we will connect to slave node without any key.**
* **Use some Ad-Hoc commands:**

1. ansible all -m ping
2. ansible --list-hosts all
3. ansible all -m command -a "uptime"
4. ansible all -m command -a "hostname"
5. [anadmin@ansible-master-node ansible]$ ansible all -m command -a "uptime" -o =>for single line O/P

Output: 10.10.11.127 | CHANGED | rc=0 | (stdout) 08:58:01 up 47 min, 2 users, load average: 0.03, 0.01, 0.00

1. To know about module ===> ansible-doc <modulename>
2. ansible –help
3. ansible all -m shell -a "ip addr" -b

**Ad-Hoc Commands:**

1. ansible-doc <modulename>
2. ansible --help
3. ansible all –list-hosts
4. ansible all -m ping
5. ansible <groupname> -m ping
6. ansible all -m command -a “uptime”
7. ansible all -m command -a “hostname”
8. ansible all -m command -a “uptime” -o
9. **To create user**===>ansible all -m user -a “name=pavan” -b
10. ansible <groupname> -m user -a “name=pavan” -b
11. if you have custom inventory/host file then use
12. ansible -i <hostfile with full path> -m user -a “name=pavan” -b
13. **To copy the file**===>ansible all -m copy -a “src=/home/ec2-user/abc.sh dest=/var/tmp/abc.sh” -b
14. ansible all -m copy -a “src=/home/ec2-user/abc.sh dest=/var/tmp/abc.sh mode=0755” -b
15. ansible all -m file -a “src=/home/ec2-user/abc.sh mode=0755” -b
16. **To make file** ===>ansible all -m file -a “path=/var/tmp/text01.txt state=touch” -b
17. ansible all -m file -a “path=/var/tmp/text01.txt state=touch mode=0755” -b
18. **for delete file**===> ansible all file -a “path=/var/tmp/text01.txt state=absent” -b
19. **To make Directory**===> ansible all -m file -a “path=/var/tmp/test state=directory” -b
20. **To know memory details**===> ansible all -m shell -a “cat /proc/meminfo | head -5”
21. **To know free memory**===> ansible all -m shell -a “free -m”
22. **To know statics of file**===> ansible all -m stat -a “path=filepath”
23. **To know IP address===>** ansible all -m shell -a "ip addr" -b
24. **To install software’s===>** ansible all -m yum -a “name=git” -b
25. ansible <group-name> -m git -a "repo=https://foo.example.org/repo.git dest=/srv/myapp version=HEAD"
26. ansible <group-name> -m **apt -a 'update\_cache=yes' -b**
27. ansible <group-name> -m **apt -a 'upgrade=dist update\_cache=yes' -b**
28. **To check CPU usage**===>ansible hakase-testing -m shell -a 'mpstat -P ALL' –become
29. **To check open ports===>**ansible hakase-testing -m shell -a 'netstat -plntu' --become

**Ad-Hoc commands to install packages in slaves by using master**

* To install “httpd/apache2”

1. ansible <group-name> -m **yum -a “name=httpd state=present” -b**
2. ansible <group-name> -m **apt -a “name=apache2 state=present” -b**
3. ansible <group-name> -m service -a "name=httpd enabled=yes state=started" -b
4. ansible <group-name> -m **service -a "name=httpd state=started" -b**
5. ansible <group-name> -m service -a "name=httpd enabled=yes state=started" -b

**note:**

state= present, started, latest, installed, absent, stopped, restarted

modules= **yum, apt, service**

* To install “git”

1. ansible <group-name> -m **apt -a "name=git state=latest" -b**
2. ansible <group-name> -m **yum -a "name=git state=latest" -b**
3. ansible <group-name> -m **apt -a "name=git state=present" -b**
4. ansible <group-name> -m **yum -a "name=git state=present" -b**

* To install “Nginx”

1. ansible <group-name> -m apt -a 'name=nginx state=latest' -b
2. ansible <group-name> -m yum -a 'name=nginx state=latest' -b
3. ansible <group-name> -m apt -a 'name=nginx state=**absent** purge=yes' -b
4. ansible <group-name> -m apt -a 'name=nginx state=absent purge=yes **autoremove=yes**' -b
5. ansible <group-name> -m service -a 'name=nginx **state=started** enabled=yes' -b
6. ansible <group-name> -m service -a 'name=nginx **state=restarted'** --become
7. ansible <group-name> -m service -a 'name=nginx **state=stopped'** –become

* To install “Mysql”

1. ansible db\_servers -m yum -a 'name=mysql-server **state=present'** -b
2. ansible web\_servers -m apt -a 'name=mysql-server **state=present'** -b

**note:**

to check status of mysql

**for ubuntu:** service mysql status

**for linux:** service mysqld status

1. for linux we need to start mysql no need for ubuntu

ansible db\_servers -m service -a 'name=mysqld enabled=yes **state=started'** -b

1. ansible web\_servers -m service -a 'name=mysql **state=stopped'** -b
2. ansible web\_servers -m service -a 'name=mysql **state=restarted'** -b

**Playbook commands:**

**Note:**

* ansible-playbook <playbook file name with path> --syntax-check
* ansible-playbook <playbook file name with path > --check
* time ansible-playbook <playbook file name with path > --check
* ansible-playbook <playbook file name with path > --list-hosts --list-tasks

1. sudo su <ansible-user>
2. cd /etc/ansible
3. sudo mkdir playbooks
4. cd /playbooks
5. sudo touch sample.yml
6. sudo vi sample.yml
7. ansible all -m setup ===> to know about ansible server
8. ansible all -m setup |grep ansible\_os\_family
9. ansible-playbook package.yml
10. ansible-playbook package.yml **--list-tags**
11. ansible-playbook package.yml **--tags 'git install'**

**sample playbook:**

---

- name: this is sample playbook to install some packages in slave nodes

hosts: all

become: true

tasks:

- name: install httpd

yum:

name: httpd

state: installed

when: ansible\_os\_family == "RedHat"

- name: start httpd

service:

name: httpd

state: restarted

when: ansible\_os\_family == "RedHat"

- name: insatll apache2

apt:

name: apache2

state: present

when: ansible\_os\_family == "Debian"

- name: start apache2

service:

name: apache2

state: started

when: ansible\_os\_family == "Debian"

- name: install git

yum:

name: git

state: present

when: ansible\_os\_family == "RedHat"

**Playbook with tags:**

---

- name: this is sample playbook to install some packages in slave nodes

hosts: all

become: true

tasks:

- name: install httpd

yum:

name: httpd

state: installed

when: ansible\_os\_family == "RedHat"

tags: install httpd

- name: start httpd

service:

name: httpd

state: restarted

when: ansible\_os\_family == "RedHat"

tags: started httpd

- name: insatll apache2

apt:

name: apache2

state: present

when: ansible\_os\_family == "Debian"

tags: install apache2

- name: start apache2

service:

name: apache2

state: started

when: ansible\_os\_family == "Debian"

tags: started apache2

- name: install git

yum:

name: git

state: present

when: ansible\_os\_family == "RedHat"

tags: git install

**note:**

1. ansible-playbook package.yml **--list-tags**
2. ansible-playbook package.yml **--tags 'git install'**

**Playbook with “lineinfile”, “handlers”, “tags”, “var”:**

---

- name: this is sample playbook to install some packages in slave nodes

hosts: all

become: true

vars:

port: 800

tasks:

- name: install httpd

yum:

name: httpd

state: present

when: ansible\_os\_family == "RedHat"

- name: start httpd

service:

name: httpd

state: started

when: ansible\_os\_family == "RedHat"

- name: changes in configuration port

lineinfile:

path: /etc/httpd/conf/httpd.conf

regexp: '^Listen'

insertafter: '^#Listen'

line: Listen {{port}}

when: ansible\_os\_family == "RedHat"

notify: restart httpd

- name: insatll apache2

apt:

name: apache2

state: present

when: ansible\_os\_family == "Debian"

- name: start apache2

service:

name: apache2

state: started

when: ansible\_os\_family == "Debian"

- name: changes in configuration port

lineinfile:

path: /etc/httpd/conf/httpd.conf

regexp: '^Listen'

insertafter: '^#Listen'

line: Listen {{port}}

when: ansible\_os\_family == "RedHat"

notify: restart httpd

- name: insatll apache2

apt:

name: apache2

state: present

when: ansible\_os\_family == "Debian"

- name: start apache2

service:

name: apache2

state: started

when: ansible\_os\_family == "Debian"

- name: changes in configuration port

lineinfile:

path: /etc/apache2/ports.conf

regexp: '^Listen'

insertafter: "^# /etc/apache2/sites-enabled/000-default.conf"

line: Listen {{port}}

when: ansible\_os\_family == "Debian"

tags: change in port

notify: restart apache2

- name: install git

yum:

name: git

state: present

when: ansible\_os\_family == "RedHat"

handlers:

- name: restart httpd

service:

name: httpd

state: restarted

- name: restart apache2

service:

name: apache2

state: restarted

**if any permission error in httpd start**

**like** Permission denied: AH00072: make\_sock: could not bind to address [::]:8000

**fix by running where httpd running:**

**Disable SELinux**

Disable SELinux temporarily

sudo setenforce 0

Restart httpd service

service httpd restart

Disable SELinux persistently (after reboot)

vi /etc/selinux/config

Add line and save

SELINUX=disabled

**Playbook for Nginx:**

---

- name: This is sample playbook for install nginx

  hosts: all

  become: true

  tasks:

  # removing httpd

   - name: removed httpd

     yum:

      name: httpd

      state: absent

     when: ansible\_os\_family == "RedHat"

  # removing httpd

   - name: install nginx

     apt:

      name: apache2

      state: absent

     when: ansible\_os\_family == "Debian"

   # installing Nginx in redhat

   - name: install nginx

     yum:

      name: nginx

      state: present

     when: ansible\_os\_family == "RedHat"

   # nginx starting stage in RedHat

   - name: start nginx

     service:

      name: nginx

      state: started

     when: ansible\_os\_family == "RedHat"

   # install nginx in Debian

   - name: installation nginx

     apt:

      name: nginx

      state: present

     when: ansible\_os\_family == "Debian"

   # nginx starting in Debian

   - name: start nginx

     service:

      name: nginx

      state: started

     when: ansible\_os\_family == "Debian"

ROLES:

1. cd /etc/ansible/roles (or) cd /etc/ansible
2. mkdir roles
3. cd roles
4. mkdir apache-git ===> role name (or) sudo ansible-galaxy init apache-git
5. cd apache-git
6. mkdir files handlers tasks vars meta templates
7. in each directory we need to create “sudo touch main.yml”

like below: [anadmin@ansible-master roles]$ tree

└── apache-git

├── defaults

│   └── main.yml

├── handlers

│   └── main.yml

├── tasks

│   └── main.yml

└── vars

└── main.yml

1. cd playbooks
2. sudo vi package.yml

---

- name: this is sample playbook to install some packages in slave nodes

hosts: all

become: true

roles:

- apache-git

1. ansible-playbook package.yml --syntax-check
2. ansible-playbook package.yml –check
3. If all ok “ansible-playbook package.yml”

Note points:

sudo amazon-linux-extras install epel -y

sudo yum-config-manager --enable epel

sudo yum -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm

10.4.1.141 ansible\_ssh\_pass=xxx ansible\_ssh\_user=user

yum install sshpass -y

**Ansible also used as IAC (infrastructure as code):**

* By using we can create AWS resources like EC2, VPC, S3 etc…
* Check below link

https://docs.ansible.com/ansible/latest/collections/amazon/aws/ec2\_module.html

* Use this command “ansible-galaxy collection install amazon.aws”