Configure ssh key-based authentication for users(sujit, devops) and sudo privileges all linux server and DNS entry.

ansible --version

[sujit@server ansible]$ cat /etc/ansible/hosts

[webservers]

webserver1.example.com

webserver1.example.com

[test]

desktop

Use Ad Hoc Commands:

----------------------

ansible all --list-hosts

ansible webservers -m command -a "uptime" -o #-a 'ARGUMENTS', --args='ARGUMENTS' -o, --one-line -m NAME, --module-name=NAME

Work with Modules:

-----------------------

ansible webservers -m yum -a "name=httpd state=present" -b -o

ansible webservers -m service -a "name=httpd state=started" -b #-b flag tells Ansible to become the privileged user and is necessary

Stop and Uninstall Apache Service:

---------------------------

Disable the web service:

[devops@control ~]$ ansible webservers -m service -a "name=httpd state=stopped" -b

Remove Apache by setting the httpd state of the yum module to absent:

[devops@control ~]$ ansible webservers -m yum -a "name=httpd state=absent" -b

Write First Playbook:

---------------------------

Create a directory to hold your playbook and change to it:

[devops@control ~]$ mkdir ~/apache\_basic

[devops@control ~]$ cd ~/apache\_basic

[devops@control ~]$ cat > install\_apache.yml <<EOF

---

- name: Install the apache web service

hosts: webservers

become: yes

tasks:

- name: install apache

yum:

name: httpd

state: present

- name: start httpd

service:

name: httpd

state: started

EOF

[devops@control ~]$ ansible-playbook --syntax-check install\_apache.yml

Run playbook:

---------------------------

[devops@control ~]$ ansible-playbook install\_apache.yml

Use Variables, Loops, and Handlers:

---------------------------

In the previous section, you created a useful, but limited, playbook. In this section you use variables, loops, and handlers to create a more powerful

playbook.

Create a playbook called main.yml in directory called apache\_intermediate and add the playbook header information:

[devops@control ~]$ mkdir ~/apache\_intermediate

[devops@control ~]$ cd ~/apache\_intermediate

[devops@control ~]$ cat > main.yml <<EOF

---

- name: Install Apache playbook

hosts: webservers

become: yes

gather\_facts: false

vars:

httpd\_packages:

- httpd

- python3-mod\_wsgi

apache\_test\_message: This is a test message

tasks:

- name: install httpd packages

yum:

name: "{{ item }}"

state: present

loop: "{{ httpd\_packages }}"

notify: restart apache service

handlers:

- name: restart apache service

service:

name: httpd

state: restarted

enabled: yes

EOF

[devops@control ~]$ ansible-playbook main.yml

Create Role

--------------------------

[devops@control ~]$ mkdir -p ~/apache\_intermediate/roles

[devops@control ~]$ cd ~/apache\_intermediate/

Use the ansible-galaxy command to initialize a new role called a\_simple\_webserver:

[devops@control ~]$ ansible-galaxy init roles/a-simple-webserver

[devops@control ~]$ tree roles/a-simple-webserver/

[devops@control ~]$ cd roles/a-simple-webserver/

[devops@control ~]$ rm -rf files tests

[devops@control ~]$ cat > defaults/main.yml <<EOF

# defaults file for a-simple-webserver

apache\_test\_message: This is a test message

EOF

[devops@control ~]$ cat > vars/main.yml <<EOF

# vars file for a-simple-webserver

httpd\_packages:

- httpd

- python3-mod\_wsgi

EOF

[devops@control ~]$ cat > handlers/main.yml <<EOF

# handlers file for a-simple-webserver

- name: restart apache service

service:

name: httpd

state: restarted

enabled: yes

EOF

[devops@control ~]$ curl http://www.opentlc.com/download/ansible\_downloads/index.html.j2 -o templates/index.html.j2

[devops@control ~]$ cat templates/index.html.j2

[devops@control ~]$ cat > tasks/main.yml <<EOF

---

# tasks file for apache-simple

- name: install httpd packages

yum:

name: "{{ item }}"

state: present

loop: "{{ httpd\_packages }}"

notify: restart apache service

- name: copy index.html

template:

src: index.html.j2

dest: /var/www/html/index.html

- name: start httpd

service:

name: httpd

state: started

enabled: yes

EOF

[devops@control ~]$ cd ~/apache\_intermediate/

[devops@control ~]$ mv main.yml main.yml.bak

[devops@control ~]$ cat > main.yml <<EOF

---

- name: This is my role-based playbook

hosts: webservers

become: yes

roles:

- a-simple-webserver

EOF

Finally, run your new main.yml playbook:

[devops@control ~]$ ansible-playbook main.yml