

Ansible:-

Installation:-

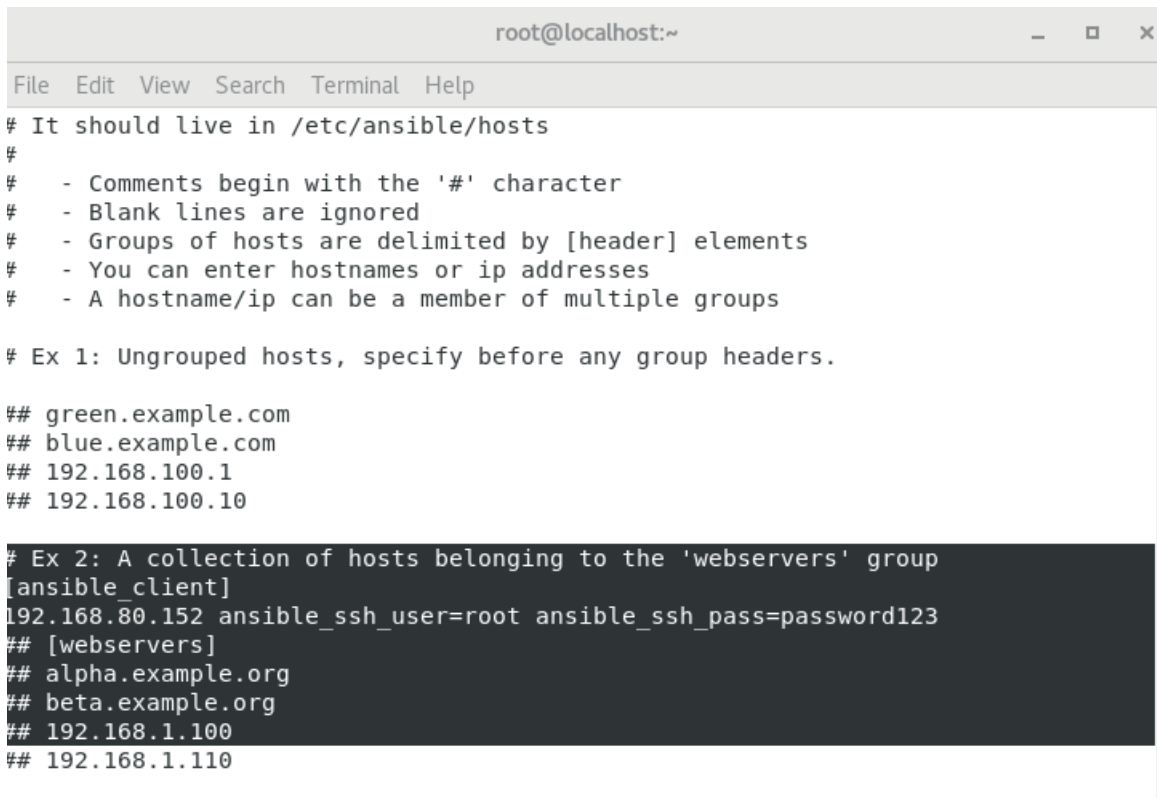
```
# yum install epel-release
```

```
# yum install ansible -y
```

```
#cat /etc/ansible/hosts
```

```
[ansible_client]
```

```
192.168.80.152 ansible_ssh_user=root ansible_ssh_pass=password123
```



```
root@localhost:~
File Edit View Search Terminal Help
# It should live in /etc/ansible/hosts
#
# - Comments begin with the '#' character
# - Blank lines are ignored
# - Groups of hosts are delimited by [header] elements
# - You can enter hostnames or ip addresses
# - A hostname/ip can be a member of multiple groups
#
# Ex 1: Ungrouped hosts, specify before any group headers.
## green.example.com
## blue.example.com
## 192.168.100.1
## 192.168.100.10
# Ex 2: A collection of hosts belonging to the 'webserver' group
[ansible_client]
192.168.80.152 ansible_ssh_user=root ansible_ssh_pass=password123
## [webserver]
## alpha.example.org
## beta.example.org
## 192.168.1.100
## 192.168.1.110
```

```
vim sample.yaml
```

```
---
```

```
- name: sample book
```

```
  hosts: ansible_client
```

```
  become: true
```

```
  tasks:
```

```
    - name: installing httpd
```

```
      yum:
```

```
        name: httpd
```

```
    state: latest
- name: running httpd

service:
    name: httpd
    state: started
- name: creating content

copy:
    content: "Congrats on Installing ansible"
    dest: /var/www/html/index.html
```

```
---
- name: sample book
  hosts: ansible_client
  become: true
  tasks:
    - name: installing httpd
      yum:
        name: httpd
        state: latest
    - name: running httpd
      service:
        name: httpd
        state: started
    - name: creating content
      copy:
        content: "Congrats on Installing ansible"
        dest: /var/www/html/index.html
```

To check syntax:-

```
# ansible-playbook sample.yml --syntax-check
```

```
playbook: sample.yml
```

```
[root@localhost ~]# ansible-playbook sample.yml --syntax-check
playbook: sample.yml
[root@localhost ~]# ansible-playbook sample.yml

PLAY [sample book] *****

TASK [Gathering Facts] *****
fatal: [192.168.80.152]: FAILED! => {"msg": "Using a SSH password instead of a key is not possible because Host Key checking is enabled and sshpass does not support this. Please add this host's fingerprint to your known_hosts file to manage this host."}

PLAY RECAP *****
192.168.80.152 : ok=0    changed=0    unreachable=0    failed=1    skipped=0    rescued=0    ignored=0
```

First take ssh of client , so server will have fingerprint of client. Then run playbook.

```
[root@localhost ~]# ansible-playbook sample.yml

PLAY [sample book] *****

TASK [Gathering Facts] *****
ok: [192.168.80.152]

TASK [installing httpd] *****
changed: [192.168.80.152]

TASK [running httpd] *****
changed: [192.168.80.152]

TASK [creating content] *****
changed: [192.168.80.152]

PLAY RECAP *****
192.168.80.152 : ok=4    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Running a Script on all client: -

```
cat calc.sh
```

```
#!/bin/bash
```

```
#read -p "Enter a number:" num1
```

```
num1=$1
```

```
#read -p "Enter a smaller number:" num2
```

```
num2=$2
```

```
#read -p "Enter an operand:" op
```

```
op="$3"
```

```
echo $op
```

```
if [[ "$op" == "+" ]]
```

```
then
```

```

    echo "$num1 + $num2 = $((num1 + num2))"
elif [[ "$op" == "-" ]]
then
    echo "$num1 - $num2 = $((num1 - num2))"
elif [[ "$op" == "*" ]] # Corrected the condition for multiplication
then
    echo "$num1 * $num2 = $((num1 * num2))"
elif [[ $op == "/" ]]
then
    echo "$num1 / $num2 = $((num1 / num2))"
else
    echo "Operator not listed"
fi

```

```

# ansible ansible_client -m script -a "calc.sh 12 12 +"
192.168.80.128 | CHANGED => {
  "changed": true,
  "rc": 0,
  "stderr": "Shared connection to 192.168.80.128 closed.\r\n",
  "stderr_lines": [
    "Shared connection to 192.168.80.128 closed."
  ],
  "stdout": "+\r\n12 + 12 = 24\r\n",
  "stdout_lines": [
    "+",
    "12 + 12 = 24"
  ]
}

```

ansible\_client → Group name

-m : module