Cloud Automation using Ansible

Ansible is used to automate your daily tasks on existing infrastructure, eliminating the time-consuming. Ansible is a free open source.

Redhat ----Ansible works only on Linux/Ubuntu OS.

Ansible is built with Python and ansible works only with Yaml.

The architecture of Linux: Ansible on the control node needs some information to successfully connect to managed hosts.

Components:

- Control node
- managed nodes.
- Inventory: This contains the IP address of the managed nodes.
- Playbooks: It is a set of instructions written in the language called Yaml.

Step 1: Install Ansible on AWS.

```
sudo amazon-linux-extras install ansible2
```

```
[root@ip-192-168-1-20 ec2-user]# sudo amazon-linux-extras install ansible2
Installing ansible
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Cleaning repos: amzn2-core amzn2extra-ansible2 amzn2extra-docker amzn2extra-kernel-5.10
22 metadata files removed
8 sqlite files removed
0 metadata files removed
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
amzn2extra-ansible2
amzn2extra-docker
amzn2extra-kernel-5.10
(1/9): amzn2-core/2/x86_64/group_gz
(2/9): amzn2-core/2/x86_64/updateinfo
(3/9): amzn2extra-docker/2/x86_64/primary_db
(4/9): amzn2extra-kernel-5.10/2/x86_64/updateinfo
```

Step 2: Adding Host for target machines(instances)

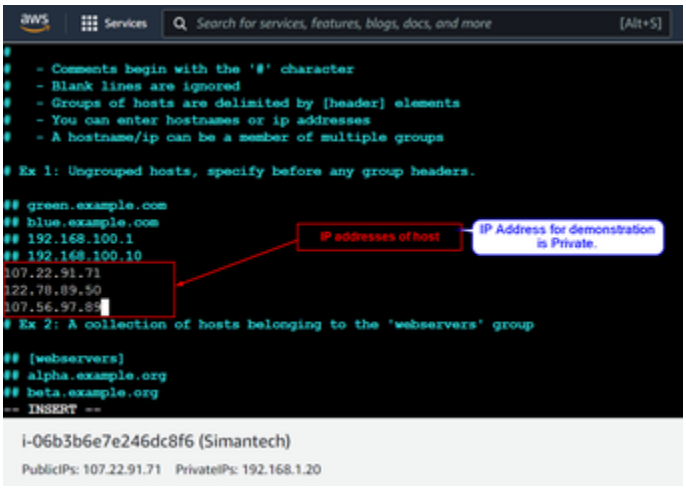
Follow the following terminals to add Ip addresses of

- Cd /etc/ansible
- edit hosts to config managed Nodes for automation of cloud

```
vim hosts
```

```
[root@ip-192-168-1-20 ansible]# ls -la
total 36
drwxr-xr-x  3 root root    51 Aug  7 11:04 .
drwxr-xr-x 82 root root  8192 Aug  7 11:04 ..
-rw-r--r--  1 root root 19985 Jul  1  2021 ansible.cfg
-rw-r--r--  1 root root  1016 Jul  1  2021 hosts
drwxr-xr-x  2 root root     6 Jul  1  2021 roles
[root@ip-192-168-1-20 ansible]#
```

Edit hosts in order to add the host IP address on which automation has to be performed.



Step 3: Creating yaml Playbooks(Script)

Ansible Playbooks offer a repeatable, re-usable, simple configuration management and multi-machine deployment system, one that is well suited to deploying complex applications. If you need to execute a task with Ansible more than once, write a playbook and put it under source control. Then you can use the playbook to push out new configuration or confirm the configuration of remote systems.

```
---
- hosts: all
  remote_user: root
  tasks:
   - name: copying the files from one location to other location
     copy:
       src: /etc/ansible/playbook.yml
       dest: /tmp/playbook.yml
       follow: yes

*****************************

---
- hosts: all
  remote_user: root
  tasks:
    - name: install the httpd
      yum: name=httpd state=present
    - name: starting the service
      service: name=httpd state=started
    - name: check the status
      shell: "systemctl status httpd"

**************************

ansible-playbook playbook.yml --extra-vars "ansible_ssh_user=root ansible_ssh_pass=password"
```

Step 4: Create a YML script.

- Vi httpd.yml
- add following script in order to install httpd package on ec2 instance.

```
---
-hosts: all(can be replaced)
 remote_user: root
 tasks:
  -name: install the httpd
  Yum: name=httpd state=present
  -name: starting the service
  service: name=httpd state=started
```

:wq!

Step 5: Password Authentication

```
Cat /etc/ss

Vi /etc/ssh/sshd_config
Systemctl restart sshd
```

Step 6: To execute ansible Playbook:

```
Ansible-playbook httpd.yml --extra-vars "ansible_ssh_user=root
ansible_ssh_pass=admin123"
```