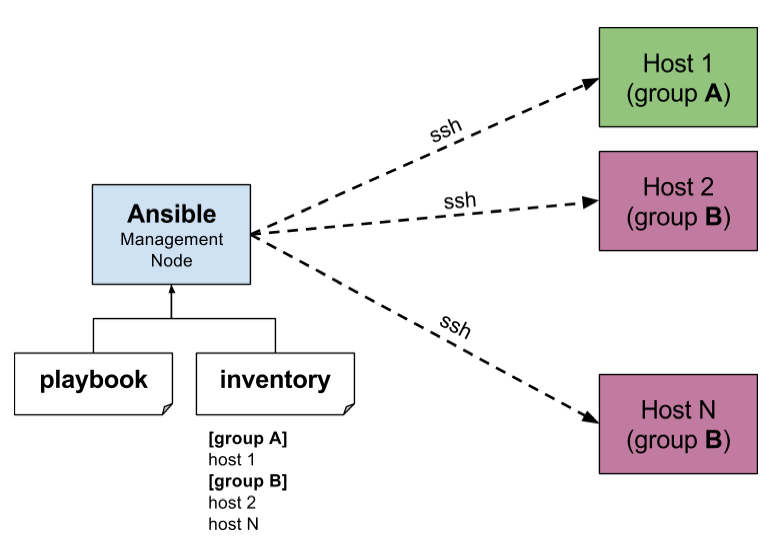
# What is Ansible

1. Ansible is a software tool that is used for **application deployment**, **updates** on workstations and servers, cloud **provisioning**, **configuration** management, intra-service **orchestration**, and nearly anything a systems administrator does on a weekly or daily basis.
2. It’s an open-source tool which enables **infrastructure as code**.

# How it works

1. Ansible management nodes have an **inventory** (server list)
2. Ansible will connect to these servers via **passwordless ssh** (use root user)
3. Ansible can execute any command/action to a single host or a group of nodes
4. Ansible can run a **playbook** (set of instruction/commands) to a single host or a group of nodes



# <3 Environment

1. ansible-server: 192.168.0.100
2. ansible-client: 192.168.0.101

# Step#1: Ansible Installation

## @Ubuntu 22.04

mahidul@mahidul:~/devops$ sudo apt install ansible

mahidul@mahidul:~/devops$ ansible --version

## @CentOS 8

[root@localhost ~]# dnf makecache

[root@localhost ~]# dnf install epel-release

[root@localhost ~]# dnf install ansible

## @Ubuntu

mahidul@ansible-server:~$ sudo apt install ansible

# Step#2: Prepare client-server

***I. Enable root user from client server/hosts***

root@mahidul:~# passwd root

root@client:~# vim /etc/ssh/sshd\_config

PermitRootLogin yes

root@mahidul:~# systemctl restart ssh

***II. Enable passwordless connection to the client-server***

root@ansible-server:~$ ssh-keygen

root@ansible-server:~$ ssh-copy-id -i /root/.ssh/id\_rsa.pub root@192.168.1.101

# <3 Configure Ansible for automated deployment

## Add hosts

[root@ansible-server ~]# vim /etc/ansible/hosts

[myservers]

192.168.1.101

## Check all hosts

root@ansible-server ~]# ansible -i /etc/ansible/hosts --list-hosts all

## Check if a host is reachable

[root@ansible-server ~]# ansible -i hosts 192.168.0.101 -m ping

## Check if a group is reachable

[root@ansible-server ~]# ansible -i hosts myservers -m ping

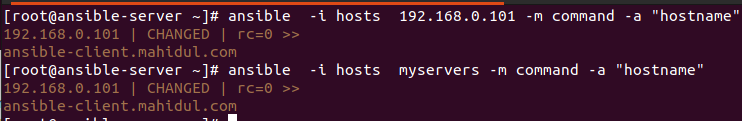


## Run a single command

We can call the servers with **IP** or **group names**.

[root@ansible-server ~]# ansible -i hosts 192.168.0.101 -m command -a "hostname"

[root@ansible-server ~]# ansible -i hosts myservers -m command -a "hostname"



## Make Configuration Changes on Node

~#ansible -m lineinfile -a "path=/etc/resolv.conf line=nameserver 8.8.4.4" demoservers

## Create a playbook (a set of instructions)

[root@ansible\_master ~]# vim /etc/ansible/hosts

[grp\_web]

192.168.0.103

Install and start nginx service for Unbuntu

---

- hosts: webservers

tasks:

- name: check\_ping

ping:

tags: checkping

- name: install apache

apt:

name: apache2

state: present

force\_apt\_get: yes

- hosts: databaseservers

tasks:

- name: check\_ping

ping:

tags: checkping

Install and start nginx service for CentOS

[root@ansible\_master ~]# vim instruction.yml

---

- hosts: grp\_web

tasks:

- name: check ping

ping:

tags: check

- name: install nginx

yum: pkg=nginx state=installed

notify: notify nginx started

tags: nginx

- name: install snmp

yum: pkg=net-snmp state=installed

tags: snmp

- name: start snmp

service: name=snmpd state=started

- name: start nginx

service: name=nginx state=started

- name: Ensure SELinux is set to enforcing mode

ansible.builtin.lineinfile:

path: /etc/selinux/config

regexp: '^SELINUX='

line: SELINUX=disabled

- name: Disable selinux

ansible.builtin.shell:

cmd: setenforce 0 # To list files under /opt directory

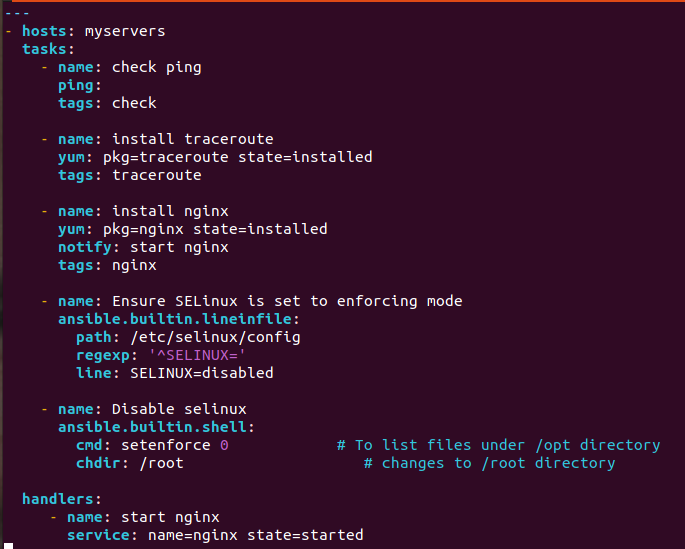
chdir: /root # changes to /root directory

handlers:

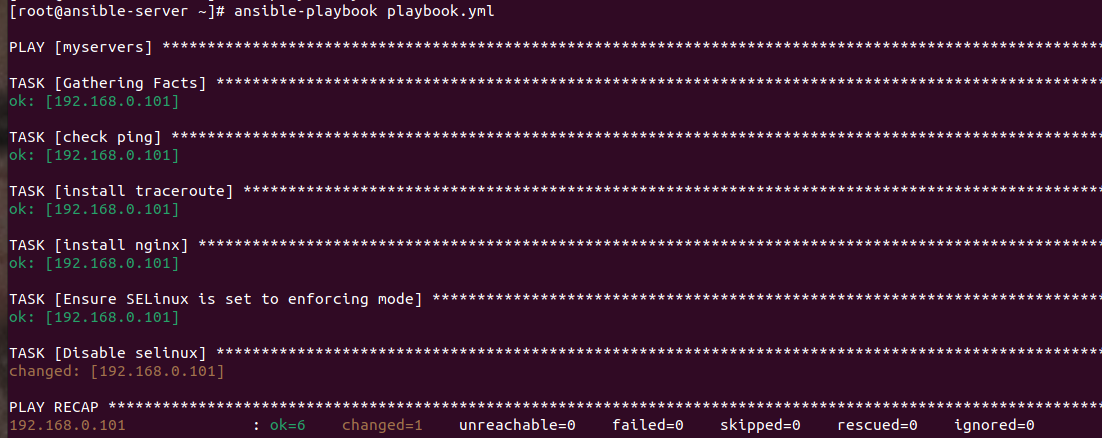
- name: notify nginx started

service: name=nginx state=restarted

[root@ansible-server ~]# ansible-playbook instruction.yml



[root@ansible-server ~]# ansible-playbook playbook.yml



## Run only a name

[root@ansible-server ~]# ansible-playbook playbook.yml --tag 'check ping'

## Run only a tag

[root@ansible-server ~]# ansible-playbook playbook.yml --tag check

# <3 Reference

https://docs.ansible.com/ansible/latest/collections/ansible/builtin/lineinfile\_module.html