**Ansible**

1. It’s a configuration management tool
2. Its widely used option for devops automation

We can automate following

* Infrastruce
* Applications Deployments
* Networks
* Containers (Docker,kubernates): instead of hell we can use Ansible for deploying apps to kubernetes
* Security
* Cloud

1. Other tools other than Ansible

* Cheft
* Puppet
* Salt stag

Out of all these Ansible is leading.

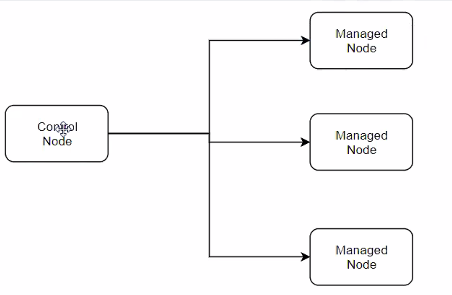
What is configuration Management?

* For ex: we want to install Docker on 1000 servers or we want to install and configure tomcat on 100 servers
* We want to apply patches on all servers we use Ansible

Why ansible?

* Open source
* Ansible is simple to learn and simple to use
* We don’t have to learn a new scripting language**.**
* **Ansible is agent less**(cheft,puppet uses agent concept)

18-11-19

Ansible:

Ansible:

A machine where we install in ansible

Its must be a non windows

It sould have a specific python2 (version2.7) or python3 (version3.5and higher ) installed

**Managed node requirements:**

Any machine which is managed by ansible

The comm b/w manged node and control node is ssh

For windows it uses “winrm”

It suppoets windows and linux as a managed node

You also need python2(version 2.6 or later

IQ)What is Ansible inventory pipe:

Ansible requires inventory file which is a txt file which contains Ip address user name, password and connection type.which is used by ansible control node for communicating with managed nodes

IQ) Ansible.cfg

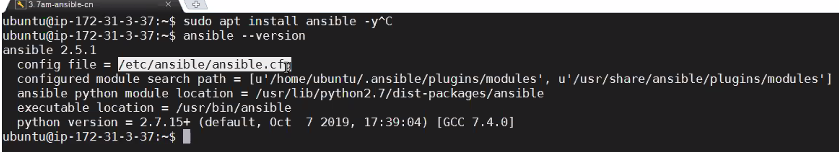
It is a configuration file through which we can change the behaviour of Ansible.

Setting up ansible control machine on ubuntu:

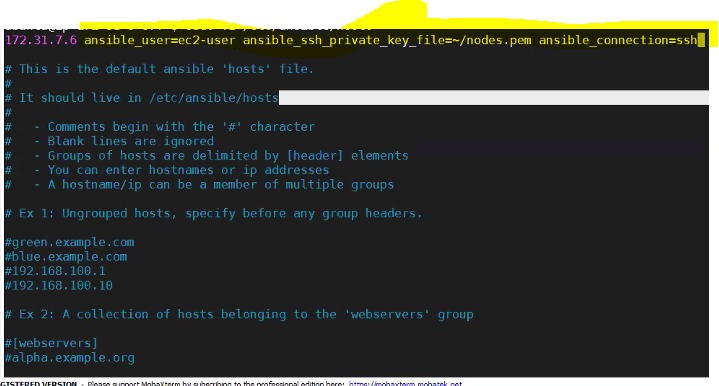
# sudo apt update -y (update the repo cache)

#sudo apt install ansible -y (to install ansible y stands for yes)

#ansible –version



Setting up managed node:

* Launch Linux machine for managed node
* make sure python version
* Add managed node detailes in inventory pipe
* #sudo vi/etc/ansible/hosts (open this file add details)
* 

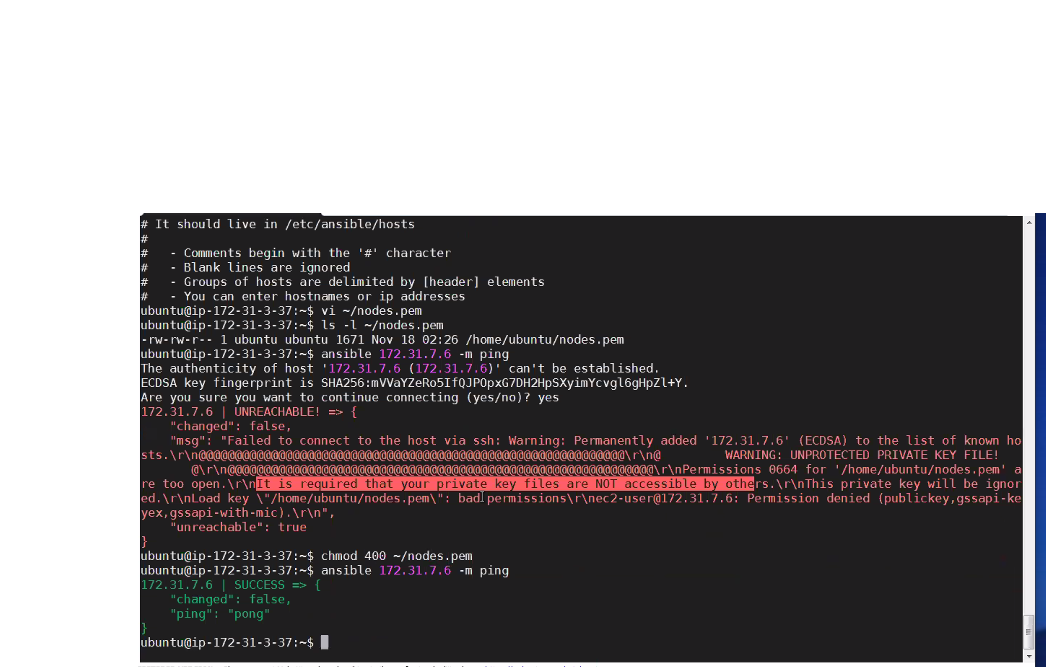
We need to make sure the dile is kept in specified location in the above command

#vi ~/nodes.pem ( paste the pem data and save)

#ls -l ~/nodes.pem

#chmod 400 ~/nodes.pem

#ansible 172.31.7.6 -m ping



Ansible Modules:

We have to use a specific module for any kind of automation for ex: (we want to install a package we should use **yum** module for linux app module for Ubuntu.

IQ) What is Adhoc command

Any ansible command which is tied upfront for performing a specific automation

Ex: ansible 172.31.7.6 -m ping

#ansible 172.31.7.6 -m yum -a ‘name=docker state =present’ --become

perform the above task with sudo



It concludes that docker is successfully installed.

Ansible is idempotent i.e running same cmd repeatedly has no side effects.

make docker to start

#ansible 172.31.7.6 -m service -a ‘name=docker state=started’ --become

#ansible 172.31.7.6 -m service -a ‘name=docker enable=yes’ --become

**Ansible playbooks:**

* **We w**rite playbooks in yaml.
* In a playbook we describe sequence of automations we want to do on managed nodes.

#vi docker.yml

--- (indicaes file isyml file)

---

Hosts: 172.31.7.6

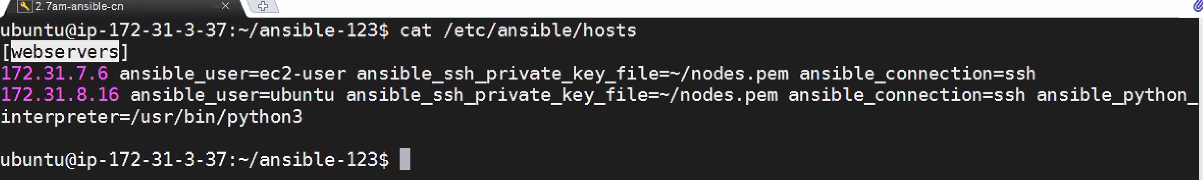
Become: yes

Tasks:

* Name: Install docker
* Yum:
* Name:docker
* State:present
* Name: start docler
* Service:
* Name:docker
* State:started
* Enabled:yes or true



#ansible -playbook docker.yml



----------------------19-11-19 (abscent)---------------------

**20-11-19**

**Ansible Raw command:**

**This Is useful and should only be done in a few cases. A common case is installing python on a system without python istalled by default.**

Running a specific node on a group:

$ Ansible webservers[0] -m ping

$ Ansible webservers[0:1] -m ping (to run nodes by range)

[webservers:vars]

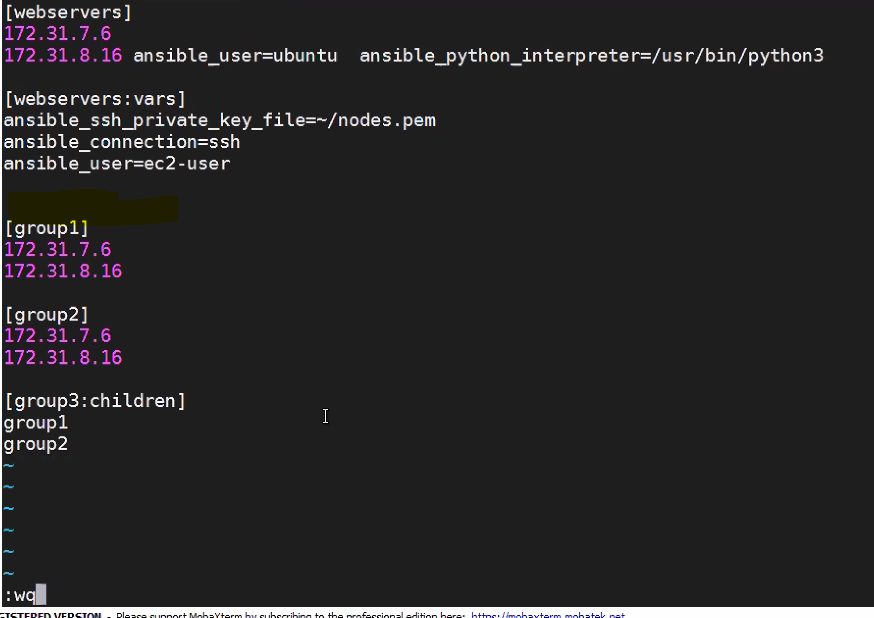
Ansible\_ssh\_private\_key\_file=~/nodes.pem

Ansible\_connection=ssh

Ansible\_user=ec2-user

**Group of groups:**

[group1]



If we want to declare variables.

db-[a:f].example.com

$ ansible webservers -I dev.inv -m ping

$ ansible webservers -i dev.inv -m ping (to ping particular inventory by using ‘-i’)

$ ansible webservers -I dev.inv --list hosts (to get lists of hosts on a particular inventory) it is helpful when doing troubleshooting.

Ex: we want to install apache on webservers.

Vi apache.webservers

Hosts: webservers

Become:yes

Tasks:

Name: Install servers

Name:httpd

Ansible -playbok -I dev.inv apache.yml

**Date: 21-11-19:**

Changing the port of apache:

* I need to find where to change port number

1. Directly editing the file

**Ansible handlers:**

Handlers are tasks, there are executed only when it is notified

For ex: we have a task to change a port number and we want to restart apache only when the port number is changed.

**Using variables in playbook:**

Vars files:

**IQ) How do you maintain secrets in Ansible:**

* Using vault -vault is used to store passwords in encrypted format.

**Ansible vault:**

To encrypt the password

$ ansible vault encrypt vars.yml 🡪file name

$ ansible-playbook --help | grep vault

$ ansible-playbook apache.yml --ask-vault-pass

2nd option): $ ansible-playbook apache.yml --vault-password-file=~/vault password

How to view encrypted files:

How to edit encrypted variable file

$ ansible-vault edit vars.yml

How to decrypt the password

$ ansible vault decrypt vars.yml

**22-11-19**

$ ansible-vault encrypt string “password” --name “git password”

After encryption of pass , put it in **vi.vars.yml file**

**Multiple vault ID and passwords:**

A vault i

**Ansible templates**:

Ansible templates are dynamic config files

Ansible uses jinja2 template module for producing dynamic configuration files.

Python uses jinja2 in dynamic web app development.

* Ex:- I want a config file or txt file with all web server ip servers

Jinja we can use for loops

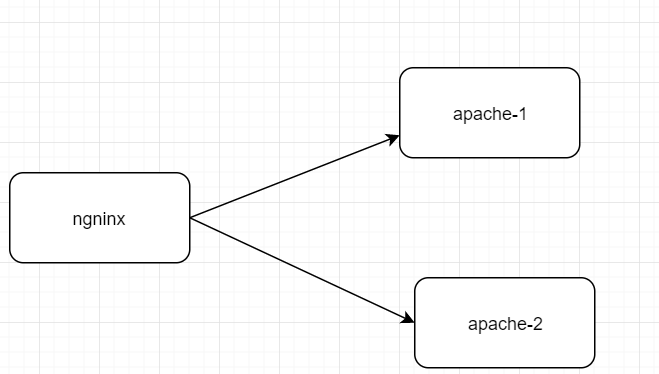
IQ) what is difference b/w template and file

* Template is dynamic config file & file is static config file

Configuring nginx load balancing:

Apache-1

Apache-2



**Writing reusable playbooks:**

We can divide playbooks & tasks in to separate file and include them wherever you want to re-use.

Ex:1 reusing playbooks – write one.yml playbook and import it in two.yml

Ex2: write three.yml which contains only tasks, import that in two.yml