**ANSIBLE**

Ansible helps in provisioning and orchestration to the server, it is the configuration management in the DevOps practise

Similar ansible tools are chef/Puppet and Docker (Containerisation process)

Basic Configuration files for Ansible,

**ansible 2.4.2.0**

**config file** = /etc/ansible/ansible.cfg

**configured module search path** = [u'/root/.ansible/plugins/modules',u'/usr/share/ansible/plugins/modules']

**ansible python module location** = /usr/lib/python2.7/site-packages/ansible

**executable location** = /usr/bin/ansible

**python version** = 2.7.14 (default, Jun 19 2018, 17:14:23) [GCC 7.3.1 20180303 (Red Hat 7.3.1-5)]

To check the version of Ansbile,

**ansbile --version**

The current version of ansible we use for the practise is 2.4.2.0(with the latest release and the fixpack). You can use apt/yum (depends on the flavour of Linux) repository tool to upgrade the ansible version to the latest (in future, if needed)

Two basic files which are considered to be most important in the ansible are “Hosts” and the “Playbooks”

Hosts is the default file which will be created while the installation of ansible, it basically contains the host address where the operations are to be executed. It can be segregated into groups (ex – Webservers/Database servers/Application Servers/Test Servers) depending on our convenience. There is no restriction for naming the group name (except the special symbols). The default location of hosts will be in the directory /etc/ansbile/hosts

Playbook is the yml file to play around with the operations. This file contains the module details, host details, user details and tasks which is to be performed. It should end with an extension <filename>. Yml. It can be executed in any location by calling the command,

**ansible-playbook <filename.yml>**

Now that we are going to see the usages and execution of ansible tool,

1. **First we will check the ssh connection for the servers in the hosts file,**

**ansible all -m ping**

Where,

* ansible – represents the call function
* all – represents the list all in the hosts file
* -m – is the flag to call the module
* Ping – is the module which we have called

This should obviously give a reply mentioning the connection being established in the server which is mentioned in the hosts file. In simple terms, a successful acknowledgment in the target server/servers

The design of packages in ansible are built in python language and all the module details can be seen under ansible documents. For instance, let’s take the above example,

Ping should give the connection established status, the same what ansible does will be defined in the ansible doc, that can be viewed by using the below command,

**ansible-doc <package name>**

Since, we take the example of ping,

**ansible-doc ping**

We will have the package details

**> PING (/usr/lib/python2.7/site-packages/ansible/modules/system/ping.py)**

By seeing the file ping.py we will know the algorithm of ping and the response for the execution of anbile ping, the response for ansbile ping in given below,

**<IP Address> | SUCCESS => {**

**"changed": false,**

**"ping": "pong"**

**}**

If incase, you find some errors in the response, you have to check for the error if the concern was for OS or ansible. In general, if you are using a fresh OS that has not been added under authorised keys, you might get into trouble while executing ping package that can be resolved by doing the following steps,

* **Goto, root directory**
* **List the hidden files and get into .ssh directory**
* **You will see the authorised\_key file**
* **Create the pub and private by using ssh-keygen**

**id\_rsa: PEM RSA private key**

**id\_rsa.pub: OpenSSH RSA public key**

* **add the “id\_rsa.pub” to the authorised\_key file**

1. **Ansible Playbook**

Installation of any products can be done easily through ansible and thereby one can do the provisioning process across the servers in their infrastructure. This can be possible through ansible playbook. Ansible playbook is simply the yml file which contains all the required details say host, task and the state of the packages. Here we take the example of installing git in all servers mentioned in hosts file

Create your playbook (. yml) in the preferred directory like the below,

**---**

**- name: install git latest version**

**hosts: all**

**become: true**

**become\_user: root**

**gather\_facts: true**

**tasks:**

**- name: "Install git"**

**package: name=git state=present**

command to run the playbook created,

**ansible-playbook <playbookname.yml>**

An important factor to be considered, if you are using the python version above 2(i.e, 3), it will throw out a fatal error stating “python2 bindings for rpm are needed for this module. python2 yum module is needed for this module". Fix for the same is change the python version to 2 as the current version of python supported by ansible is python2

**ansible-python-interpreter: /usr/bin/python2**

The playbook created above is just an example to show the installation of git using ansible and the state in the task section will enable and disable the package(present/absent) by changing the parameters