**ANSIBLE DOCUMENTATION**

**Pre-Requisites :**

* Python should be installed in “Ansible Control Server” as well as “Node Servers”.

**Ansible:**

* Ansible is a “Configuration Management Tool”. It is used to automate the deployment.
* Ansible is a Push Model configuration server.
* In Ansible we have “Control server” and multiple “Nodes servers”.
* We can connect Control server to Node servers through SSH.
* Ansible is a Light Weight Control server.
* In Ansible we write Playbooks. These are in the form of YAML.
* There is no programming is required for Ansible.

**Control Server:**

* Ansible Control Server has to be a Linux Server.

**YAML:**

* YAML is a “Data language”

1. **We need to create an USER (Ansible User).**
2. **Give the Sudo access to Ansible User.**
3. **From Ansible Control Server if we need to Login into systems with key based authentication.**

**Environmental setup for Ansible:**

* First we have to take one Ansible Control Server. For this we need to login to AWS console and Launch the Linux instance because Control server should be LINUX.
* Launch the instance
* Add the User in Ansible Control Server(ACS). Goto root (sudo -i)

$ adduser <Enter username>

$ passwd <Enter password> (You should manually create the password using this command in RedHatLinux. In Ubuntu you will get

the option like “Enter New password”)

$ su <username>

* Switch to root user

$ visudo (in this file write user details like below)

# Allow members of group sudo to execute any command

%sudo ALL=(ALL:ALL) ALL

ansible ALL=(ALL:ALL) NOPASSWD:ALL

* Then switch to Ansible User.

$ sudo apt-get update

* Goto the location etc/ssh/sshd\_config

$ vi etc/ssh/sshd\_config

* Modify the sshd\_config file   
  passwordauthentication yes
* Restart the service

$ sudo service shhd restart

* After restart exit from the user.
* Then login to ansible user

$ ssh ansible@<DNS name of instance>

* **Installing Ansible on Control Server**
* From this site you can install Ansible on different environments (https://docs.ansible.com/ansible/latest/installation\_guide/intro\_installation.html)

$ sudo apt-get update

$ sudo apt-get install software-properties-common

$ sudo apt-add-repository --yes --update ppa:ansible/ansible

$ sudo apt-get install ansible

$ python –version

$ ansible --version

* **Inventory:** Where all the nodes are maintained is called as Inventory file. When you install Ansible by deafault we will get Inventory file. (etc/ansible/hosts)
* We will Add Node servers ip-addresses in the hosts file.

$ sudo vi hosts

<node server 1 Private ip-address/Private DNS name>

<node server 2 Private ip-address/Private DNS name >

<node server 3 Private ip-address/Private DNS name >

<node server 4 Private ip-address/Private DNS name >

Ex:

192.168.1.1

* If we are using different environments like Group of Ubuntu machines and Group of RedHat machines. We should write inventory file like below

[ubuntu]

<node server 1 Private ip-address/Private DNS name>

<node server 2 Private ip-address/Private DNS name >

[redhat]

<node server 3 Private ip-address/Private DNS name >

* **Configurations in the Ansible push via two ways:**
* Command Line (Adhoc commands) (ansible)
* YAML (Palybooks) (ansible-playbook)
* **Setting Authentication**
* First we need to generate key on master node i.e Ansible Control Server( goto this path /etc/ansible)

$ ssh-keygen

* With the above command the ssh\_key will generate in the following path (/home/ansible/.ssh/id\_rsa)
* After generating key we need to copy the key to a directory where exactly we wanted to copy on Node servers.

$ ssh\_copy\_id <username>@<Private IP or Private DNS name>

$ ssh\_copy­\_id ansible@localhost

* After this ssh to the localhost i.e to machine ($ssh localhost)
* Then ping the machine

$ ansible –m ping all

* **Launching Node server (Node-1)(aws instance as ubuntu)**
* Logging to Node-1 server

$ sudo -i

$ adduser ansible

Enter new unix password:

Retype new unix password:

* Switch to root user

$ visudo (in this file write user details like below)

# Allow members of group sudo to execute any command

%sudo ALL=(ALL:ALL) ALL

ansible ALL=(ALL:ALL) NOPASSWD:ALL

* Goto the location etc/ssh/sshd\_config

$ vi etc/ssh/sshd\_config

* Modify the sshd\_config file   
  passwordauthentication yes
* Restart the service

$ sudo service sshd restart

* After restart exit from the user.
* Then switch to Ansible User.

$ sudo apt-get update

* **Ansible-Playbook:**
* **If I want to update packages on Node servers we have to write a playbook.**
* **Steps to write playbook:**

1. Update the Ubuntu packages
2. Find the module which can do update

* Goto ACS and create a file called update.yaml or update.yml

$ vi update.yml

---

- hosts: all

become: yes

tasks:

- name: update packages on ubuntu

apt:

name: git

update\_cache: yes

state: present

* name: install tree

apt:

name:tree

state: present

* To execute the playbook

$ ansible-playbook update.yml

* **Writing playbook for installing Apache, Updating apache and installing PHP:**

---

- hosts: all

become: yes

tasks:

- name: update packages & install apache

apt:

name: apache2

update\_cache: yes

state: present

* name: restart apache after installation

service:

name:apache2

enabled: yes

state: restarted

* name: Install PHP packages

apt:

name: “{{ item }}”

state: present

with\_items:

- php

- libapache2-mod-php

- php-mcrypt

- php-mysql

* name: restart apache after php

service:

name:apache2

enabled: yes

state: restarted

* To execute the playbook

$ ansible-playbook update.yml

* **Handlers:**

Handler is used to call the repeated tasks in the playbook. Instead of writing tasks again and again we will use Handlers. In handlers we have “notify” directive, by using notify we call the handlers.

* Ex : For the above playbook we will write playbook using handlers to simplify the playbook.
* We are restarting apache again&again to avoid repeated code we will use handlers like below.

---

- hosts: all

become: yes

tasks:

- name: update packages & install apache

apt:

name: apache2

update\_cache: yes

state: present

notify:

* restart apache
* name: Install PHP packages

apt:

name: “{{ item }}”

state: present

with\_items:

- php

- libapache2-mod-php

- php-mcrypt

- php-mysql

notify:

* restart apache

handlers:

* name: restart apache

service:

name:apache2

enabled: yes

state: restarted