Ansible Installation

Ansible is an open source automation platform. Ansible can help you with configuration management, application deployment, task automation.

# Prerequisites

1. AWS EC2 instance

# Installation steps:

Add a EPEL (Extra Packages for Enterprise Linux) third party repository to get packages for Ansible

rpm -Uvh https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm

**Install Ansible**

yum install ansible -y

Check Ansible version

ansible --version

Create a new user for ansible administration & grant admin access to user (Master and Slave)

useradd ansadmin

passwd ansadmin

# below command addes ansadmin to sudoers file. But strongly recommended to use "visudo" command if you are aware vi or nano editor.

echo "ansadmin ALL=(ALL) ALL" >> /etc/sudoers

# sed command replaces "PasswordAuthentication no to yes" without editing file

sed -ie 's/PasswordAuthentication no/PasswordAuthentication yes/' /etc/ssh/sshd\_config

Login as a ansadmin user on master and generate ssh key (Master)

ssh-keygen

Copy keys onto all ansible client nodes (Master)

ssh-copy-id <target-server>

Update target servers information on /etc/ansible/hosts file (Master)

echo "<target server IP>" > /etc/ansible/hosts

Run ansible command as ansadmin user it should be successful (Master)

ansible all -m ping

2 Ways ansible can perform configuration.

1) Ad-hoc commands

2) Playbooks

**Important modules in ansible**

1) Command - This module is used for executing basic linux commands on managed nodes.

2) Shell - This module is used to execute commands which involved redirection and piping and

to execute shell scripts on managed nodes.

3) Ping -- This module is used to check if the remote server is pingable or not.

4) User -- This module is used for user management like create user, setting password, assign home directory etc.

5) Copy -- This module is used to copy the files and folders from controller to managed nodes

6) Fetch -- This module is used to copy files and folder from managed nodes to controller

7) File -- This module is used for creating or deleting files and folders on managed nodes.

8) State -- Used to capture detailed information about files and folders present in managed nodes.

9) Debug -- Used to display output of any module

10) Apt -- Used for performing package management on managed nodes ie installing softwares / upgrading repositories etc. It works on ubuntu, debain flavours of linux.

11) Yum -- similar to apt module. It works on Red hat linux, centos etc

12) Git -- used to perform git version controlling on managed nodes

13) Replace -- This is used to replace specific text in configuration file with some other text.

14) Service -- used for starting / stoping / restarting services on managed nodes.

15) Include -- Used for calling child play books from parent play book

16) Uri -- useful in checking if remote url is reachable or not.

17) Docker\_container -- used to execute docker commands related to container management on managed nodes

18) Docker\_image -- used to execute commands related to docker images on managed nodes.

19) Docker\_login -- used to login to docker hub from managed nodes.

20) Setup -- used to capturing system information related to the managed nodes.

# 1) Ad-hoc Commands

Ex: ansible -i /etc/ansible/hosts -m command -a 'free' (inventory, module, argument)

ansible all -m user -a 'name=one passsword=one' -b (become, for higher privileges on

managed nodes)

**Notes:**

Ad-hoc commands are capable of working only on one module and one set of arguments.

When we want to perform complex configuration management activities,

Ad-hoc commands will be difficult to manage.

In such scenarios, we use play books.

Play book is combination of plays.

Each play is designed to do some activity on the managed nodes.

These plays are created to work on single host or a group of hosts or all the hosts.

The main advantage of play books is reusability.

Play books are created using yaml files.

# 2) Playbooks

Ex 1: Ansible playbook for install git and clone the remote repository

$ mkdir playbooks

$ cd playbooks

$ vim playbook.yml

INSERT mode

---

- name: Install git and clone a remote repository

hosts: all

tasks:

- name: Install git

apt:

name: git

state: present

update\_cache: yes

- name: clone remote git repository

git:

repo: https://github.com/shivardy06/test.git

dest: /home/ ansadmin /newgit

...

To check the syntax:

$ ansible-playbook playbook.yml --syntax-check

( Do not use tab when creating yml file )

To run the playbook

$ ansible-playbook playbook.yml -b