**ANSIBLE-AUTOMATION TOOL**









**Installation:-**

**To install ansible:-**

**# yum install epel-release –y**

**# yum install vim wget unzip -y**

**# yum install ansible –y**

**To check the ansible version:-**



**ansible python module location**

**/usr/lib/python2.7/site-packages/ansible**

**Ansible Default Configuration File:-**

**/etc/ansible/ansible.cfg**

**Ansible Default host inventory File:-**

**etc/ansible/hosts**

**To list all the modules :-**

**# ansible-doc –list**



**To check the description about a module (ex:- command, shell etc..)**

**# ansible-doc command**

**# ansible-doc shell**

**# ansible-doc user**

**# ansible-doc file**

**# ansible-doc copy**

**# ansible-doc setup**

**# ansible-doc service**

**# ansible-doc debug**

**# ansible-doc yum**





**Ansible Configuration File:-**





**[defaults] :-**

Most of the settings in the configuration file are grouped under this section.

**[inventory]:-**

Location of the Inventory File Path.

**[privilege\_escalation]:-**

This section contains settings for defining how operations which require escalated privileges will be executed on managed hosts.

**[paramiko\_connection], [ssh\_connection], [persistent\_connection], [accelerate]**

These sections contain settings for optimizing connections to managed hosts.

**[selinux]:-**

**This section contains settings for defining how SElinux interactions will be configured.**

**[colors]**

**This options forces color mode even when running without a TTY:**

**[diff]:-**

**This controls the cutoff point (in bytes) on --diff for files.**



**AD-HOC COMMAND in Ansible:-**







**To Check ip address of the hosts**

# ansible 192.168.2.1 -i myinventory --list-hosts

**To Check hosts list in the lab group.**  
#ansible lab -i myinventory --list-hosts

**To check hostname,group,ip address**

**# ansible labhost1.example.com,test,192.168.2.2 -i myinventory --list-hosts**

**To List all hosts in inventory file**

**# ansible all -i myinventory --list-hosts**

**To list all hosts in inventory using wildcard entry (\*)**

# ansible "\*" -i myinventory --list-hosts

# ansible "\*.exmple.com" -i myinventory --list-hosts

# ansible "192.168.2.\*" -i myinventory --list-hosts

# ansible "datacenter\*” -i myinventory --list-hosts

# **ansible "lab:datacenter\*” -i myinventory --list-hosts**



**# ansible "lab:&datacenter\*” -i myinventory --list-hosts**



**# ansible "lab:!test2.example.com\*” -i myinventory --list-hosts**



**# ansible "all:!datacenter1” -i myinventory --list-hosts**

**Dynamic Inventories:-**







**Implementing Playbook:-**





**Playbooks:-**

**Vim my-first-playbook.yaml**

---

- name: my first playbook...

hosts: all

remote\_user: ansible

become: yes

tasks:

- name: install httpd...

yum:

name: httpd

state: latest

- name: enable httpd service...

service:

name: httpd

state: started

enabled: true

- name: create index file with content...

copy:

content: "Hello my first playbook"

dest: /var/www/html/index.html

...

# ansible-playbook –syntax-check <playbookname> // To check syntax of playbook .

# ansible-playbook --check <playbookname> // To dry run playbook

# ansible-playbook <playbook\_name> // Actuall run playbook



**Write an ansible playbook usin block.**

**vim block.yaml**

---

- name: my first playbook...

hosts: all

remote\_user: ansible

become: yes

tasks:

- block:

- name: install httpd...

yum:

name: httpd

state: latest

- name: install firewalld...

yum:

name: firewalld

state: latest

- block:

- name: enable httpd service...

service:

name: httpd

state: started

enabled: true

- name: enable firewalld service...

service:

name: firewalld

state: started

enabled: true

- block:

- name: enable firewall rules...

firewalld:

service: http

permanent: true

state: enabled

immediate: yes

- block:

- name: create index file with content...

copy:

content: "Hello playbook"

dest: /var/www/html/index.html

- name: test url for above setup...

hosts: localhost

become: yes

remote\_user: ansible

tasks:

- name: Connect to internet...

uri:

url: [http://node1.example.com](http://node1.example.com/)

status\_code: 200

register: output

- debug: var=output

