**Quick Start**

**Installation and Setup**

* Clientless install
* No daemon to be run
* Works completely SSH
* You need ansible installed, you can use that box for ansible playbooks

**Installation and Setup**

*yum install epel-release*

*yum update*

*yum install git python python-devel python-pip openssl ansible*

*ansible --version*

**Configuration Options in Ansible**

vi /etc/ansible/ansible.cfg

inventory = /etc/ansible/hosts

sudo\_user = root

**Ansible Nodes are defined by Names - are written in /etc/ansible/hosts**



**Giving Access to ansible on Remote Nodes by creating user with sudo privilege on the node**

On ubuntu Node

* adduser ansible
* visudo



* su - ansible
* sudo apt-get update

On centos Node

* adduser ansible
* passwd ansible
* visudo



* su - ansible
* sudo yum update

**To avoid password, setup ssh key exchange**

**On Central Node [doshihardikn6.mylabserver.com]**

* su ansible -
* ssh-keygen (leave defaults when prompted)
* ssh-copy-id [ansible@doshihardikn6.mylabserver.com](mailto:ansible@doshihardikn6.mylabserver.com)
* ssh-copy-id [ansible@doshihardikn5.mylabserver.com](mailto:ansible@doshihardikn5.mylabserver.com)
* Test your password less connection
  + **ssh doshihardikn5.mylabserver.com** as ansible user
  + **ssh doshihardikn6.mylabserver.com** as ansible user
* Must Copy ID command on localhost to make sure, ssh is passwordless on localhost as well

**Ansible Documentation**

* [www.ansible.com](http://www.ansible.com)
* Ansible is owned by Redhat
* RedHat certification
* Ansible Tower
  + Reporting
  + Graphing
  + Enterprise Support
* [www.ansible.com](http://www.ansible.com)
  + Go to resources
  + Click docs
  + Click on Ansible Documentation under Ansible tab
* Inventory
  + Range of IPs
  + Groups
  + Servers
* Module Index
* Ansible with AWS Course
* Ansible with Detail for other modules
* Good community
* YAML Format

**Running Ansible Commands**

* Executed following ansible module against all groups
  + ansible **all** **-m** **ping**
  + Reply in json by Ping module as pong
* List Command for all groups for files under /home/ansible directory
  + ansible all **-a “ls -al /home/ansible”**
* To run commands as root
  + ansible all **-s** -a “cat /var/log/messages”
* To run commands for a specific group
  + ansible **local** **-s** -a “cat /var/log/messages”
* Copy using module and ansible command on All nodes
  + ansible **all** **-m copy -a “src=test.txt dest=/tmp/test.txt”**
  + Reply in json
* Installing packages with a module
  + ansible ubuntu **-s** **-m apt -a “name=elinks state=present”**
  + Reply with json
* Remove packages with a module
  + ansible ubuntu **-s** **-m apt -a “name=elinks state=absent”**
  + Reply with Json
* Add a user
  + ansible centos -s -m user -a “name=test”
  + Reply with json
* Remove a user
  + ansible centos -s -m user -a “name=test **state=absent**”
  + Reply with json
  + Will not remove user directory until a specific attribute specified

**Playbook Structure with YAML**

* Start of file with Comment on what playbook does

--- #This is a structural YAML test

* Specifying hosts to be run again
  + **- hosts: centos**
* Running playbook as a specific user

**remote\_user: ansible**

* Becoming a root user

**become: yes**

**become\_method: sudo**

* Connection Type [Paramico, Local and SSH]

**connection: ssh**

* Gather\_Facts

**gather\_facts: yes**

* + Getting information about Operating System, what is installed about node
  + Against 1000s node, make it node, it will slow down marginally

* Variables

**vars:**

**username: myuser**

* Major sections are delineated by **DASH (-)**

**tasks:**

* **- name: Install HTTPD server on CentOS 7 Nodes**

**yum:**

[These are parameters of a module]

**name: httpd**

**state: latest**

[Notifies Handler named “Startservice”, if this task name is executed]

**notify:**

* **- startservice**

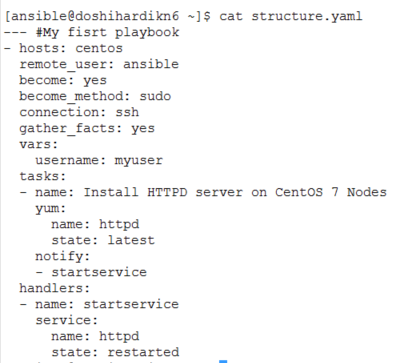
**handlers:**

* **- name: startservice**

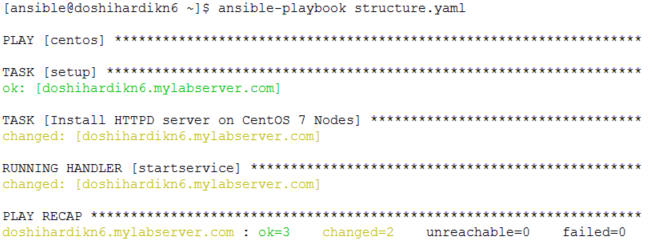
**service:**

**name: httpd**

**state: restarted**



* **Running a playbook without options**
  + ansible-playbook.yaml structure.yaml



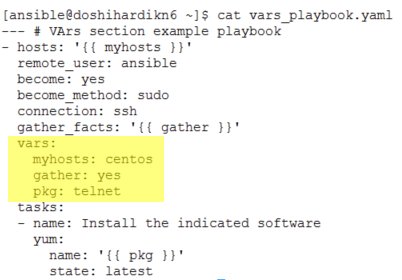
**Gathering Facts**

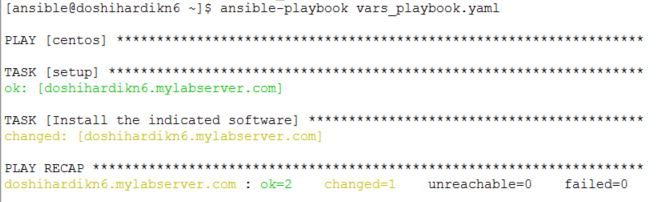
* To list all hosts using ansible command
  + ansible all --list-hosts
* Gathering facts about hosts of specific host using Setup Module and Ansible Command
  + ansible centos -m setup
* Filtering the out of the gathered facts - We can use wildcards, if we need
  + ansible centos -m setup -a ‘filter=\*ipv4\*’
  + ansible centos -m setup -a ‘filter=\*default\*’
* How to know, what to filter
  + Create an output file by following
  + ansible centos -m setup --tree factsdirectory
    - It will create [If does not exist] directory and create files for each hosts with facts
  + We can use these files for filter and can be used in variables in side playbooks

**VARIABLES IN THE PLAYBOOK**

**Using Vars Section**

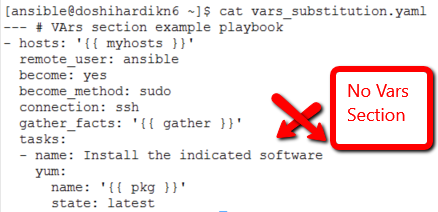
* Variables in the **vars** section
* Variables that you can pass at runtime
* **‘{{ variablename }}’ - Syntax**
* Although vars section comes after hosts, gather\_facts, you can still use variables for those sections as well.

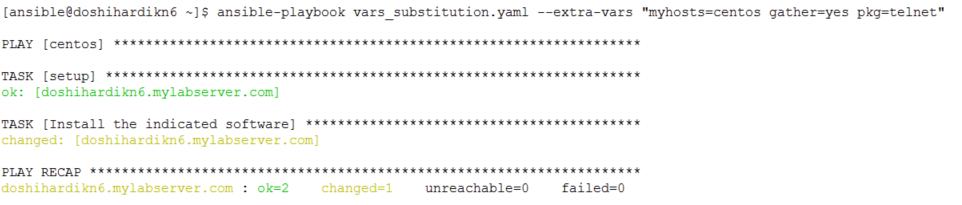




**Using Variable Substitution**

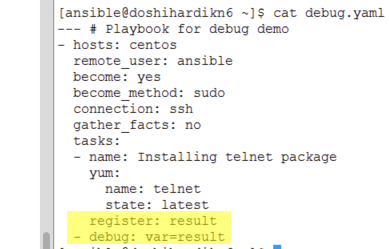
ansible-playbook vars\_substitution.yaml **--extra-vars "myhosts=centos gather=yes pkg=telnet"**



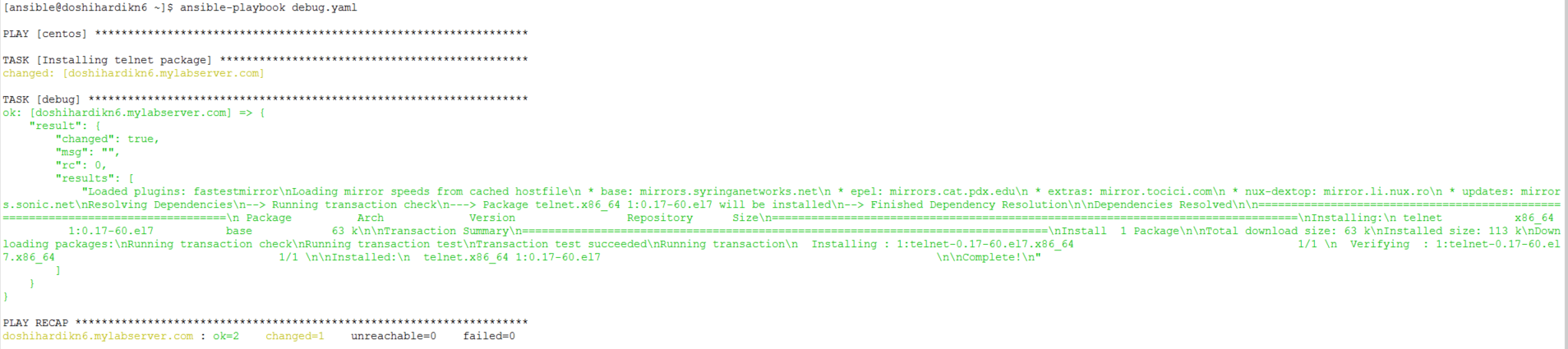


**DEBUG STATEMENT**

* Register output attribute of the task
  + register: result



* Output the result
  + debug: var=result **(After playbook execution it will show JSON output returned by module as shown below)**



* **Use Cases**
  + Details about what exactly happened on the server.
  + Script to scan through this output and format it in tabular format or add it to database field and use for reporting.

Detailed exception and error for troubleshooting

**NOTIFICATIONS AND HANDLERS**

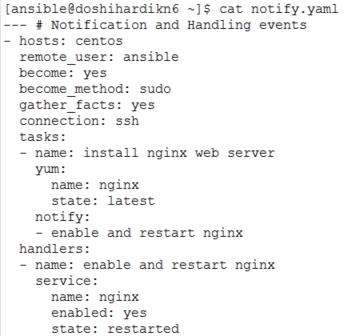
* Execute tasks with a different order or conditional run of specific task
* **Example**

**Multiple Tasks**

* + Install service using one task
  + Enable and Restart service with one more task
* If you keep Enable and Restart as task, it will always restart Nginx Server.

**Notify and Handler**

* Install service using one task
  + Notify only when Install executed
  + Handler will be called from Notify
    - Enable and Restart is called only when Service is installed



***First Run when Installation Happened***



***Next Run when Installation Task did not Execute***

