**Ansible Tutorial Points**

**1st Tutorial Getting started with Ansible**

* The purpose of Ansible is to provision servers.
* Ansible Server will connect to all the server to issue commands for provisioning.
* Ansible is installed on the Ansible server and make connection to the server via SSH and provisioning it.

**2nd Tutorial : Ansible & SSH Overview & Setup**

* ssh-keygen -t id\_ed25519 -C "Ansible key"
* cd /home
* cd balaji
* cd .ssh file key will exists
* ssh-copy-id -i id\_ed25519 subha@192.168.1.5
* ansible -all -m ping

**3rd Tutorial : Getting started with Ansible 03 - Setting up the Git Repository**

the git hub respository

Pull down the Respository from git hub respository and make changes and push up the changes to github.

**Create an user name for git hub**

* Copy the ssh key from ansible system and configure it in github
* git clone "github address" so that file from the github will be moved to the linux system.
* git config --global user.name "vbalajikumaran"
* git config --global user.email "vbalajikumaran@yahoo.co.in"
* cat ~/.gitconfig

**Git Commands**

* git status
* git diff <filename>
* git add <filename>
* git commit -m "comments" All uncommited changes.
* git push origin main

**4th Tutorial Running command**

* sudo apt update
* sudo apt install ansible
* ansible all --key-file ~/.ssh/ansible\_key -i inventory -m ping
* ansible all --list-hosts
* ansible -all -m gather\_hosts
* ansible all -m gather\_facts --limit subha@192.168.1.5

Creating config.cfg

[defaults]

inventory = inventory

private\_key\_file = ~/.ssh/ansible\_key

**5th Tutorial Running elevated commands**

* sudo-apt update : ansible all -m apt -a update\_cache=true --become --ask-become-pass
* installing vim-nox application : ansible all -m apt -a name=vim-nox --become --ask-become-pass (in client system run which vim-nox ,apt-search vim-nox)
* intallingtmuxapplicaiton :ansible all -m apt -a name=tmux --become --ask-become-pass

(in client system run which tmux ,apt-search tmux)

* in client system we can also see the history of installation

cd /var/log/apt/history.log

**updating all the upgrade in all the server**

* Server system: ansible all -m apt -a "upgrade=dist" --become --ask-become-pass
* client system: sudo apt dist-upgrade -- it should be all application are upgraded

**6th Tutorial Working with Playbook**

* 2 playbook file create install\_apache.ymlremove\_apache.yml
* ansible-playbook --ask-become-pass remove\_apache.yml

**7th Tutorial Working with When commands**

Differentiate playbook by hosts in terms of which distribution is running. (Ubuntu , Debian or Red hat)

cat /etc/os-release -- To check the server details

When command usage

* when ansible\_distribution == "Debian"
* when ansible\_distribution in ["Debian","Ubuntu"]
* when ansible\_distribution == "Ubuntu" and ansible\_distribution\_version = "8.2"

ansible all -m gather\_facts --limit subha@192.168.1.5

Centos http server will not run automatically following steps need to do

* systemctl status httpd
* sudo systemctl start httpd
* sudo firewall-cmd --add-port=80/tcp output will be --success
* run the ipaddress on the browser

8th Tutorial : Improving Playbook (Reality not going to be done)

9th Tutorial : Targeting specific Hosts

10th Tutorial: Handling Tags

Tags: Basically adding meta data on the playbook so that you can go ahead only the place we want to test which is going to make testing your playbook more easier.

For identify List of Tags in the playbook :

ansible-playbook --list-tags install\_apache.yml

For Executing specific tag :

ansible-playbook --tags db --ask-become-pass install\_apache.yml

For Executing group of tags

ansible-playbook --tags "apache,db" --ask-become-pass install\_apache.yml