**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
* sudosystemctl 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

**11th Tutorial : Copy and Managing Files**

Default ansible checks the file in file directory

Ls -l /usr/local/bin

Installing zip

Installing the zip file from the site and copying it in to Client system.

Copying single file

Copying multiple file

Copying directories

Writing contents in remote hosts.

Changing directory 4 (Read) 2(write) 1 (execute)

chmod 777 files/\*.txt

sudochownbalaji:balaji tes\*.\*

**12th Tutorial : Service Module**

12th Tutorial : Service Module

service --status all + active - active ? not known

service apache2 status

systemctl status apache2

started , stopped, restarted, reloaded , enabled

**13th Tutorial : User and Bootstraps**

cat /etc/passwd in client system to check most recently user added

creating sudoer\_anisble under files folder under ansible server system

in Client system : ls-l /etc/sudoers.d

after running ssh key and sudoers file successfully in playbook from client system checking connection using ansible user

[defaults]

* invetory = inventory
* private\_key\_file=~/.ssh/ansible\_key
* remote\_user=ansible

Add an user to the system

Add the sshkey

Add the sudoers file to automate the sudo aspect

**14. Handling Roles**

1. Update Respository Index apt\_when.yml

2. User management adding new user, sudoersfile

3. Unzip the file ad extrction Installing Terraform form the site

4. install\_apache Apache2, php , open java

5. starting the services

Roles : Repo\_update, User Creation , install\_app, install\_zip and Service

**15. Host variables and handlers**

Using Host Variables, given us additional functionalities. we can able to use playbook and task book even better

**16. Templates.**

nano /etc/ssh/ssh\_config - Configuration file. each distribution we have a seperate file.

copy this file to create\_users under roles/create\_user/templates

cp /etc/ssh/ssh\_config sshd\_config\_ubuntu.j2

.j2 is the default extension for ansible.

Add AllowUsers {{ ssh\_user }}

update ssh\_users in host\_vars each host file

ssh\_users: ansible

ssh\_template\_file :sshd\_config\_ubuntu.j2

under host\_vars in ansible sever system : cp subha@192.168.1.5.yml balaji@192.168.1.4.yml

17 Ansible vault.

. before file is the hidden file in linux

ls -l .vaultkey.txt

chmod 600 .vaultkey.txt

ansible-vault encrypt file1.txt

ansible-vault decrypt file1.txt

ansible-vault edit file1.txt

ansible-vault view file1.txt

Encrypt : ansible-vault encrypt --vault-password-file .vaultkey.txt file1.txt

Decrypt : ansible-vault decrypt --vault-password-file .vaultkey.txt file1.txt

View: : ansible-vault view --vault-password-file .vaultkey.txt file1.txt

Password change : ansible-vault rekey file1.txt --vault-password-file .vaultkey.txt