Labs - * Ansible

Friday, November 26, 2021 06:56 AM

```
ansible thbs_solution -m apt -a 'name=apache2 state=present' -i
myInventory.txt -b --ask-pass
```

```
 ansible testgroup  -m file -a 'path=/etc/hosts/'
```

```
- name: Remove file (delete file)
  file:
    path: /etc/foo.txt
    state: absent

- name: Recursively remove directory
  file:
    path: /etc/foo
    state: absent


- name: Install the latest version of Apache
  yum:
    name: httpd
    state: latest

- name: Install apache httpd  (state=present is optional)
  apt:
    name: apache2
    state: present
```

```
Ansible thbs_solution -m apt -a 'name=apache2 state:present'
```

Create inventory file

```
host0 ansible_host=192.168.33.10 ansible_user=root
host1 ansible_host=192.168.33.11 ansible_user=root
```

```
ansible -m ping all -i step-01/hosts
```

Adhoc Commands

Create inventory file

```
host0 ansible_host=192.168.33.10 ansible_user=root
```

```
ansible -i step-02/hosts -m shell -a 'uname -a' host0
```

```
ansible thbs_solutions -m shell -a 'uname -a'  -i step-02/hosts
```

```
ansible -i step-02/hosts -m copy -a 'src=/etc/hosts dest=/tmp/'host0
```

```
Ansible prod  -m copy -a 'src=/etc/hosts dest=/tmp
```

Ansible prod -m copy -a 'src=/etc/hosts dest=/tmp

Many hosts, same command

Lets say we want to get some facts about the node, and, for instance, know which Ubuntu version we have deployed on nodes, it's pretty easy:

```
ansible -i step-02/hosts -m shell -a 'grep DISTRIB_RELEASE /etc/lsb-release' all
```

all is a shortcut meaning 'all hosts found in inventory file'. It would return:

Many more facts

```
ansible -i step-02/hosts -m setup host0
```

we wanted more information (ip addresses, RAM size, etc...). The solution comes from another really handy module (weirdly) called setup: it specializes in node's *facts* gathering.

Transferring file to many servers/machines
$ Ansible abc -m copy -a "src = /etc/yum.conf dest = /tmp/yum.conf"

Creating a new directory
$ Ansible abc -m file -a "dest = /path/user1/new mode = 777 owner = user1 group =
user1 state = directory"

Deleting whole directory and files
$ Ansible abc -m file -a "dest = /path/user1/new state = absent"

Managing Packages

The Ad-hoc commands are available for yum and apt. Following are some Ad-hoc commands using
yum.

The following command checks if yum package is installed or not, but does not update it.

$ Ansible abc -m yum -a "name = demo-tomcat-1 state = present"

----

1. Creating file by adhoc command

Ansible -m file -a 'path=/tmp/surekha.txt state=present' all

----

1. Delete file by adhoc command

Ansible -m file -a 'path=/tmp/surekha.txt state=absent' all

------------- Creation of playbook ---

Playbook

```
---
- name: "create directory and add file inside"
  hosts: all
  become: YES
  tasks:
    - name: "create directory"
      file:
        path: "/tmp/surekha"
        state: directory
    - name: "copy file inside surekha"
      copy:
        src: "/tmp/config.txt"
        dest: "/tmp/surekha"
#paly no 2
- name: "update systme and install and start apache server"
  host: all
  vars:
    server_name: apache2
  tasks:
    # ansible [hostgorup] -m [module] -a ["name=value"]
    # ansible all -m apt -a 'name=apache2 state=present'
    - name: "Install {{server_name}} httpd  (state=present is optional)"
      apt: #build in module by ansible
        name: {{server_name}} #arg
        state: present #arg
    # git clone  https://github.com/sshelake25/thlabs-git.git
    # ansible all -m apt -a 'name=git state=present'
    # ansible all -m git -a 'repo=https://github.com/sshelake25/thlabs-git.git dest=/tmp/checkout' -u
userName -b -K
    - name: Git checkout my project repo
      ansible.builtin.git:
        repo: "https://github.com/sshelake25/thlabs-git.git"
        dest: /tmp/checkout
```