

```
#vi playbooks/pb1.yml
```

```
- hosts: all
```

```
remote_user: root
```

```
tasks:
```

```
- name: copy files
```

```
  copy:
```

```
    src: /root/sample.txt
```

```
    dest: /tmp/sample.txt
```

```
- name: msg display
```

```
  debug:
```

```
    msg: Files are Copied!
```

```
#ansible-playbook playbooks/pb1.yml
```

#vi playbooks/pb2.yml

- hosts: ubnt

remote_user: root

tasks:

- name: install apache

apt: name=apache2 state=latest

- name: install tomcat

apt: name=tomcat7 state=latest

- hosts: cent

tasks:

- name: install httpd

yum: name=httpd state=latest

#ansible-playbook playbooks/pb2.yml

Handlers: Running Operations On Change

- hosts: ubnt

tasks:

- name: install apache

apt: name=apache2 state=latest

notify:

- start apache

- name: install tomcat

apt: name=tomcat7 state=latest

notify:

- start tomcat

handlers:

- name: start apache

service: name=apache2 state=started

- name: start tomcat

service: name=tomcat7 state=started

When Condition

```
# ansible all -m setup -a "filter=ansible_distribution"
```

```
# ansible all -m setup -a "filter=ansible_pkg_mgr"
```

- hosts: all

become: true

tasks:

- name: install apache

apt: name=apache2 state=latest

when: ansible_distribution == "Ubuntu"

- name: install httpd

yum: name=httpd state=latest

when: ansible_distribution == "CentOS"

Ansible Variables

a Variable is Element which can hold a specific value.
Variable names should be letters, numbers, and underscores.
Variables should always start with a letter.

Ex:

```
- hosts: all
  become: true
  vars:
    pack1: bzip2
    pack2: wget
  tasks:
    - name: Install Package {{pack1}}
      yum: name="{{pack1}}" state=latest
      when: ansible_distribution=="CentOS"

    - name: Install Package {{pack2}}
      apt: name="{{pack2}}" state=latest
      when: ansible_distribution=="Ubuntu"
```

var_prompt

```
- hosts: centos
  become: true
  vars_prompt:
    - name: pack_name
      prompt: Give Package Name
      private: no
  tasks:
    - name: Install Package {{pack_name}}
      yum:
        name: "{{pack_name}}"
        state: latest
```

Ansible loops

- Ansible loops can do many things in one task, such as create a lot of users, copy a set of files ,install a lot of packages.

Ex:

- hosts: all

tasks:

- name: copy {{item}}

copy:

src: /root/files/{{item}}

dest: /tmp/{{item}}

with_items:

- abc1.txt

- abc2.txt

- abc3.txt

To add multiple users using Ansible Loops:

- hosts: all
- become: true
- tasks:
 - name: add group
 - group: name="sales" state=present
 - name: Add user {{item}}
 - user: name={{item}} groups="sales"
 - with_items:
 - user1
 - user2
 - user3