Ansible playbook can make changes to the list of servers in quick time. Ansible’s setup module is able to identify the host’s types, OS distributions, and many other facts. For an example, You got a request from a client to install “Apache” package on all the Linux hosts where you have a mix of “Debian” and “Red Hat” variants. As you aware that “Debian(apt)” and “RedHat (yum)” package management will differ from each other. In such a situation, you could gather the facts (setup module) and pass to the playbook with conditions. We will also see that how to ignore the errors and continue with next tasks.

Let’s write a single playbook to install “apache” package on both distributions.

**Environment:**

* Ansible Server –**ansible-server**
* Remote hosts – **uaans69 , gpfslinapp1**

1. Login to Ansible server and view the ad-hoc inventory.  (If you do not have one, just add the remote hosts in the file)

[linadm@ansible-server automation]$ cat lin-servers.1

gpfslinapp1

uaans69

[linadm@ansible-server automation]$

2. Create a playbook to install apache on both distributions.

[linadm@ansible-server automation]$ cat ua\_http\_install.yaml

---

- hosts: all

become: yes

tasks:

- name: Install Apache on Ubuntu

apt: name=apache2 state=present

- name: Install Apache on Red Hat

yum: name=httpd state=present

[linadm@ansible-server automation]$

3. Let’s run the playbook to see the errors.

[linadm@ansible-server automation]$ ansible-playbook -i lin-servers.1 ua\_http\_install.yaml

PLAY [all] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TASK [Gathering Facts] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [gpfslinapp1]

ok: [uaans69]

TASK [Install Apache on Ubuntu] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

fatal: [gpfslinapp1]: FAILED! => {"changed": false, "cmd": "apt-get update", "msg": "[Errno 2] No such file or directory", "rc": 2}

fatal: [uaans69]: FAILED! => {"changed": false, "cmd": "apt-get update", "msg": "[Errno 2] No such file or directory", "rc": 2}

PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

gpfslinapp1 : ok=1 changed=0 unreachable=0 failed=1

uaans69 : ok=1 changed=0 unreachable=0 failed=1

[linadm@ansible-server automation]$

Playbook failed at the first task and job terminated. It didn’t try for the RedHat hosts to complete the task.

**Ignore Errors and Continue:**

1. To ignore the errors, ansible provides the option called “ignore\_errors: True” parameter. Let’s update the playbook like below.

[linadm@ansible-server automation]$ cat ua\_http\_install.yaml

---

- hosts: all

become: yes

tasks:

- name: Install Apache on Ubuntu

apt: name=apache2 state=present

ignore\_errors: True

- name: Install Apache on Red Hat

yum: name=httpd state=present

ignore\_errors: True

[linadm@ansible-server automation]$

2. Re-run the playbook and look for errors.

[linadm@ansible-server automation]$ ansible-playbook -i lin-servers.1 ua\_http\_install.yaml

PLAY [all] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TASK [Gathering Facts] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [gpfslinapp1]

ok: [uaans69]

TASK [Install Apache on Ubuntu] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

fatal: [gpfslinapp1]: FAILED! => {"changed": false, "cmd": "apt-get update", "msg": "[Errno 2] No such file or directory", "rc": 2}

...ignoring

fatal: [uaans69]: FAILED! => {"changed": false, "cmd": "apt-get update", "msg": "[Errno 2] No such file or directory", "rc": 2}

...ignoring

TASK [Install Apache on Red Hat] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [gpfslinapp1]

ok: [uaans69]

PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

gpfslinapp1 : ok=2 changed=1 unreachable=0 failed=0

uaans69 : ok=2 changed=1 unreachable=0 failed=0

[linadm@ansible-server automation]$

The Job has been completed successfully but we can see that apt commands are tried to execute on Redhat servers and failed.

Ansible playbook can make changes to the list of servers in quick time. Ansible’s setup module is able to identify the host’s types, OS distributions, and many other facts. For an example, You got a request from a client to install “Apache” package on all the Linux hosts where you have a mix of “Debian” and “Red Hat” variants. As you aware that “Debian(apt)” and “RedHat (yum)” package management will differ from each other. In such a situation, you could gather the facts (setup module) and pass to the playbook with conditions. We will also see that how to ignore the errors and continue with next tasks.

Let’s write a single playbook to install “apache” package on both distributions.

**Environment:**

* Ansible Server –**ansible-server**
* Remote hosts – **uaans69 , gpfslinapp1**

1. Login to Ansible server and view the ad-hoc inventory.  (If you do not have one, just add the remote hosts in the file)

[linadm@ansible-server automation]$ cat lin-servers.1

gpfslinapp1

uaans69

[linadm@ansible-server automation]$

2. Create a playbook to install apache on both distributions.

[linadm@ansible-server automation]$ cat ua\_http\_install.yaml

---

- hosts: all

become: yes

tasks:

- name: Install Apache on Ubuntu

apt: name=apache2 state=present

- name: Install Apache on Red Hat

yum: name=httpd state=present

[linadm@ansible-server automation]$

3. Let’s run the playbook to see the errors.

[linadm@ansible-server automation]$ ansible-playbook -i lin-servers.1 ua\_http\_install.yaml

PLAY [all] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TASK [Gathering Facts] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [gpfslinapp1]

ok: [uaans69]

TASK [Install Apache on Ubuntu] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

fatal: [gpfslinapp1]: FAILED! => {"changed": false, "cmd": "apt-get update", "msg": "[Errno 2] No such file or directory", "rc": 2}

fatal: [uaans69]: FAILED! => {"changed": false, "cmd": "apt-get update", "msg": "[Errno 2] No such file or directory", "rc": 2}

PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

gpfslinapp1 : ok=1 changed=0 unreachable=0 failed=1

uaans69 : ok=1 changed=0 unreachable=0 failed=1

[linadm@ansible-server automation]$

Playbook failed at the first task and job terminated. It didn’t try for the RedHat hosts to complete the task.

**Ignore Errors and Continue:**

1. To ignore the errors, ansible provides the option called “ignore\_errors: True” parameter. Let’s update the playbook like below.

[linadm@ansible-server automation]$ cat ua\_http\_install.yaml

---

- hosts: all

become: yes

tasks:

- name: Install Apache on Ubuntu

apt: name=apache2 state=present

ignore\_errors: True

- name: Install Apache on Red Hat

yum: name=httpd state=present

ignore\_errors: True

[linadm@ansible-server automation]$

2. Re-run the playbook and look for errors.

[linadm@ansible-server automation]$ ansible-playbook -i lin-servers.1 ua\_http\_install.yaml

PLAY [all] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TASK [Gathering Facts] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [gpfslinapp1]

ok: [uaans69]

TASK [Install Apache on Ubuntu] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

fatal: [gpfslinapp1]: FAILED! => {"changed": false, "cmd": "apt-get update", "msg": "[Errno 2] No such file or directory", "rc": 2}

...ignoring

fatal: [uaans69]: FAILED! => {"changed": false, "cmd": "apt-get update", "msg": "[Errno 2] No such file or directory", "rc": 2}

...ignoring

TASK [Install Apache on Red Hat] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [gpfslinapp1]

ok: [uaans69]

PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

gpfslinapp1 : ok=2 changed=1 unreachable=0 failed=0

uaans69 : ok=2 changed=1 unreachable=0 failed=0

[linadm@ansible-server automation]$

The Job has been completed successfully but we can see that apt commands are tried to execute on Redhat servers and failed.

Let’s add a conditional check on the playbook to avoid those errors.

### ****Using Facts on Playbook:****

Update the playbook with ansible variable using a conditional check. “ansible\_os\_family” is one of the ansible variables from “setup” module. By default, ansible playbook gathers facts and then executes the tasks.

[linadm@ansible-server automation]$ cat ua\_http\_install.yaml

---

- hosts: all

become: yes

tasks:

- name: Install Apache on Ubuntu

apt: name=apache2 state=present

when: ansible\_os\_family == "Debian"

ignore\_errors: True

- name: Install Apache on Red Hat

yum: name=httpd state=present

when: ansible\_os\_family == "RedHat"

ignore\_errors: True

[linadm@ansible-server automation]$

2. Let’s run the playbook again.

[linadm@ansible-server automation]$ ansible-playbook -i lin-servers.1 ua\_http\_install.yaml

PLAY [all] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TASK [Gathering Facts] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [gpfslinapp1]

ok: [uaans69]

TASK [Install Apache on Ubuntu] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

skipping: [uaans69]

skipping: [gpfslinapp1]

TASK [Install Apache on Red Hat] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [gpfslinapp1]

ok: [uaans69]

PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

gpfslinapp1 : ok=2 changed=0 unreachable=0 failed=0

uaans69 : ok=2 changed=0 unreachable=0 failed=0

[linadm@ansible-server automation]$