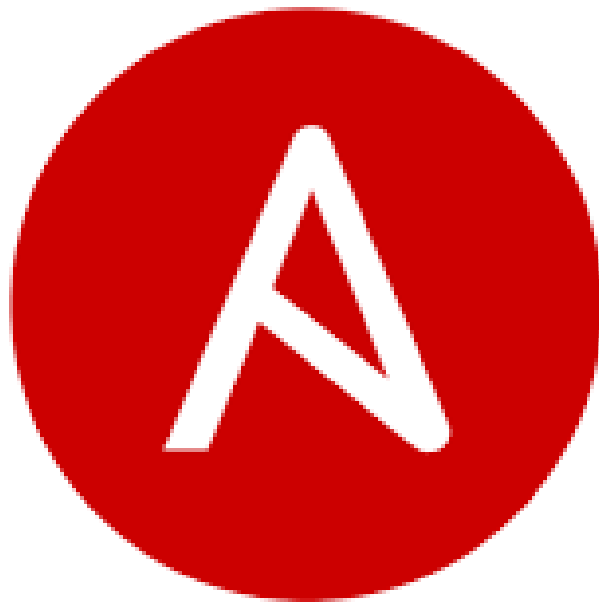


Create a dynamic Ansible playbook for deploying a webpage in any type of OS



RED HAT®
ANSIBLE®
Automation

Task Description

📖 14.3 Create an Ansible Playbook which will dynamically load the variable file named same as OS_name and just by using the variable names we can Configure our target node.(Note: No need to use when keyword here.)

First let's see little bit about the main part of our task that is ansible fact and variable.

Ansible Facts and Variable :

With **Ansible** you can retrieve or discover certain variables containing information about your remote systems or about **Ansible** itself. Variables related to remote systems are called **facts**. With **facts**, you can use the behavior or state of one system as configuration on other systems.

Ansible uses **variables** to manage differences between systems. With **Ansible**, you can execute tasks and playbooks on multiple different systems with a single command. To represent the variations among those different systems, you can create **variables** with standard YAML syntax, including lists and dictionaries.

Pre-requisite :

For doing this practical, you have to first install ansible and configure the control node.

We require Three vm's for this task — RHEL, Ubuntu and anyone for Controller Node. For ubuntu we use aws instance.

Instances | EC2 Management Console

ap-south-1.console.aws.amazon.com/ec2/v2/home?region=ap-south-1#Instances:instanceState=running

Search for services, features, marketplace products, and docs [Alt+S]

Naila Anudeep Mumbai Support

Instances (1/1) Info

Filter instances

Instance state: running Clear filters

Name	Instance ID	Instance state	Instan...	Status ch...	Alar...	Availab...	Public IPv4 DNS	Public IPv4 ...	Elastic IP
UbuntuOS	i-0c7735cba6fcb3cda	Running	t2.micro	2/2 checks	No...	ap-south-1a	ec2-65-0-76-2.ap...	65.0.76.2	-

Instance: i-0c7735cba6fcb3cda (UbuntuOS)

Details Security Networking Storage Status checks Monitoring Tags

Instance summary Info

Instance ID i-0c7735cba6fcb3cda (UbuntuOS)	Public IPv4 address 65.0.76.2 open address	Private IPv4 addresses 172.31.33.148
Instance state Running	Public IPv4 DNS ec2-65-0-76-2.ap-south-1.compute.amazonaws.com open address	Private IPv4 DNS ip-172-31-33-148.ap-south-1.compute.internal
Instance type t2.micro	Elastic IP addresses -	VPC ID vpc-02ef6e98dabb3242d
AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. Learn more	IAM Role -	Subnet ID subnet-0ecad70c1783cb8f6

Feedback English (US) © 2008 - 2021, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

Configure the ansible.cfg file.

root@ansible_controller:/etc/ansible

```
[defaults]
inventory = /etc/ansible/inventory.txt
host_key_checking = False
remote_user = ubuntu
private_key_file = /etc/ansible/ansiblekey.pem
roles_path = /etc/ansible/roles
deprecation_warnings = False
[privilege_escalation]
become = true
become_method = sudo
become_user = root
become_ask_pass = false
~
```

Configure the inventory file.

```
root@ansible_controller/etc/ansible
[redhat]
192.168.99.129 ansible_ssh_user=root ansible_ssh_pass=redhat ansible_connection=ssh
[ubuntu]
65.0.76.2 ansible_ssh_user=ubuntu ansible_ssh_private_key_file=ansiblekey.pem ansible_connection=ssh
```

Check the hosts are connected successfully or not using:

ansible all --list-hosts

ansible all -m ping

ansible all -a id

```
root@ansible_controller/etc/ansible
[root@ansible_controller ansible]# ansible all --list-hosts
hosts (2):
  192.168.99.129
  65.0.76.2
[root@ansible_controller ansible]# ansible all -m ping
65.0.76.2 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
192.168.99.129 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/libexec/platform-python"
  },
  "changed": false,
  "ping": "pong"
}
[root@ansible_controller ansible]# ansible all -a id
192.168.99.129 | CHANGED | rc=0 >>
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
65.0.76.2 | CHANGED | rc=0 >>
uid=0(root) gid=0(root) groups=0(root)
[root@ansible_controller ansible]#
```

Here our managed nodes are connected with control node successfully.

Check the facts of managed node.

“ ansible <manage_node IP> -m setup”

```
"ansible_distribution": "RedHat",  
"ansible_distribution_file_parsed": true,  
"ansible_distribution_file_path": "/etc/redhat-release",  
"ansible_distribution_file_search_string": "Red Hat",  
"ansible_distribution_file_variety": "RedHat",  
"ansible_distribution_major_version": "8",  
"ansible_distribution_release": "Ootpa",  
"ansible_distribution_version": "8.0",  
"ansible_dns": {  
  "nameservers": [  
    "49.205.171.194",  
    "49.207.34.210"  
  ]  
},
```

```
},  
"ansible_distribution": "Ubuntu",  
"ansible_distribution_file_parsed": true,  
"ansible_distribution_file_path": "/etc/os-release",  
"ansible_distribution_file_variety": "Debian",  
"ansible_distribution_major_version": "20",  
"ansible_distribution_release": "focal",  
"ansible_distribution_version": "20.04",  
"ansible_dns": {  
  "nameservers": [  
    "127.0.0.53"  
  ]  
},
```

So, “ ansible_distribution” and “ ansible_distribution_major_version” facts use here to know about OS name and its version so that we can create a dynamic playbook for deploying the webpage without using any condition.

Now, we are creating separate playbook for each managed node.

So let's create the playbook for our two manage node that is for RedHat-8 and Ubuntu-20.

Ubuntu-20.yml

```
root@ansible_controller:/etc/ansible
```

```
package_name: apache2
service_name: apache2
doc_root: /var/www/html
```

RedHat-8.yml

```
root@ansible_controller:/etc/ansible
```

```
package_name: httpd
service_name: httpd
doc_root: /var/www/html
```

creating the webpage and copy it to manage node.

task14_html.j2

```
root@ansible_controller:/etc/ansible
```

```
[root@ansible_controller ansible]# cat task14.html.j2
<h1 align="center"> THIS IS WEBPAGE FOR {{ ansible_facts['distribution'] }} </h1>
[root@ansible_controller ansible]#
```

Create the main playbook:

i.e Dynamic.yml

root@ansible_controller/etc/ansible

```
- hosts: all
vars_files:
  - "{{ ansible_facts['distribution'] }}-{{ ansible_facts['distribution_major_version'] }}.yaml"
tasks:
  - name: install webserver software
    package:
      name: "{{ package_name }}"
      state: latest
  - name: copy webpage
    template:
      src: task14.html.j2
      dest: "{{ doc_root }}/task14_3.html"
  - name: Start Web server service
    service:
      name: "{{ service_name }}"
      state: started
      enabled: yes
```

Run this playbook using command :

“ansible-playbook <playbook_name.yml> ”

root@ansible_controller/etc/ansible

[root@ansible_controller ansible]# ansible-playbook Dynamic.yml

```
PLAY [all] *****

TASK [Gathering Facts] *****
ok: [65.0.76.2]
ok: [192.168.99.129]

TASK [install webserver software] *****
ok: [65.0.76.2]

changed: [192.168.99.129]

TASK [copy webpage] *****
ok: [65.0.76.2]
changed: [192.168.99.129]

TASK [Start Web server service] *****
ok: [65.0.76.2]
changed: [192.168.99.129]

PLAY RECAP *****
192.168.99.129      : ok=4    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
65.0.76.2         : ok=4    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[root@ansible_controller ansible]#
```

Checking The Setup

ubuntu@ip-172-31-33-148: ~

```
ubuntu@ip-172-31-33-148:~$ systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2021-03-09 06:58:16 UTC; 16min ago
     Docs: https://httpd.apache.org/docs/2.4/
    Main PID: 3496 (apache2)
      Tasks: 55 (limit: 1160)
   Memory: 5.4M
    CGroup: /system.slice/apache2.service
            └─3496 /usr/sbin/apache2 -k start
              └─3498 /usr/sbin/apache2 -k start
                └─3499 /usr/sbin/apache2 -k start

Mar 09 06:58:16 ip-172-31-33-148 systemd[1]: Starting The Apache HTTP Server...
Mar 09 06:58:16 ip-172-31-33-148 systemd[1]: Started The Apache HTTP Server.
ubuntu@ip-172-31-33-148:~$
```

```
[root@MN1 ~]# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
   Active: active (running) since Tue 2021-03-09 02:10:45 EST; 4min 27s ago
     Docs: man:httpd.service(8)
    Main PID: 2007 (httpd)
   Status: "Total requests: 2; Idle/Busy workers 100/0; Requests/sec: 0.00772; Bytes served/sec: 62"
      Tasks: 278 (limit: 1520)
   Memory: 31.6M
    CGroup: /system.slice/httpd.service
            └─2007 /usr/sbin/httpd -DFOREGROUND
              └─2008 /usr/sbin/httpd -DFOREGROUND
                └─2009 /usr/sbin/httpd -DFOREGROUND
                  └─2010 /usr/sbin/httpd -DFOREGROUND
                    └─2011 /usr/sbin/httpd -DFOREGROUND
                      └─2245 /usr/sbin/httpd -DFOREGROUND

Mar 09 02:10:43 MN1 systemd[1]: Starting The Apache HTTP Server...
Mar 09 02:10:44 MN1 httpd[2007]: AH00558: httpd: Could not reliably determine the server's fully qualified domain name, setting 'ServerName' to ''
Mar 09 02:10:45 MN1 httpd[2007]: Server configured, listening on: port 8080, port 80
Mar 09 02:10:45 MN1 systemd[1]: Started The Apache HTTP Server.
lines 1-20/20 (END)
```

Instances | EC2 Management Co x 192.168.99.129/task14_3.html x 65.0.76.2/task14_3.html x +

← → ↻ Not secure | 65.0.76.2/task14_3.html ☆ 🌐 ⚙️ 👤 ⋮

THIS IS WEBPAGE FOR Ubuntu

Instances | EC2 Management Co x 192.168.99.129/task14_3.html x +

← → ↻ Not secure | 192.168.99.129/task14_3.html ☆ 🌐 ⚙️ 👤 ⋮

THIS IS WEBPAGE FOR RedHat

GitHub link: <https://github.com/Anuddeeph/Dynamic-Ansible-Playbook.git>

Conclusion:

we have learned how to load variable files according to OS distribution dynamically in the ansible-playbook using the ansible facts concept.

*using such a concept we have solved the issue of different names of the same software in different OS without using **when** keyword concept. now if we have a lot of different OS we can easily install packages or configure something according to OS specifications.*