# Lab: Managing Variables

**Introduction:**

Variable in playbooks is very similar to using variables in any programming language. It helps you to use and assign a value to a variable and use that anywhere in the playbook. One can put conditions around the value of the variables and accordingly use them in the playbook.

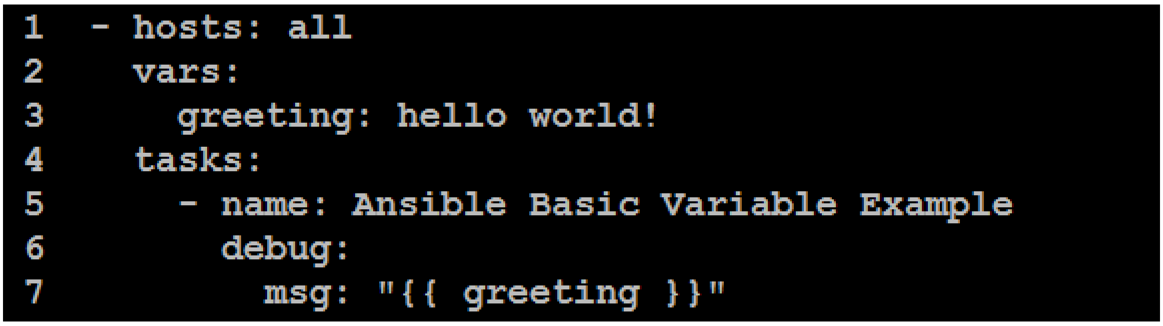
**Objectives:**

- **Creating Variables with Playbook**

## 1. Creating Variables with Playbook

**1.1** Let’s create a file name **vars.yml.**

**1.2** In the below example greeting variable is substituted by the hello world! When the playbook simply prints the message hello world! When executed.

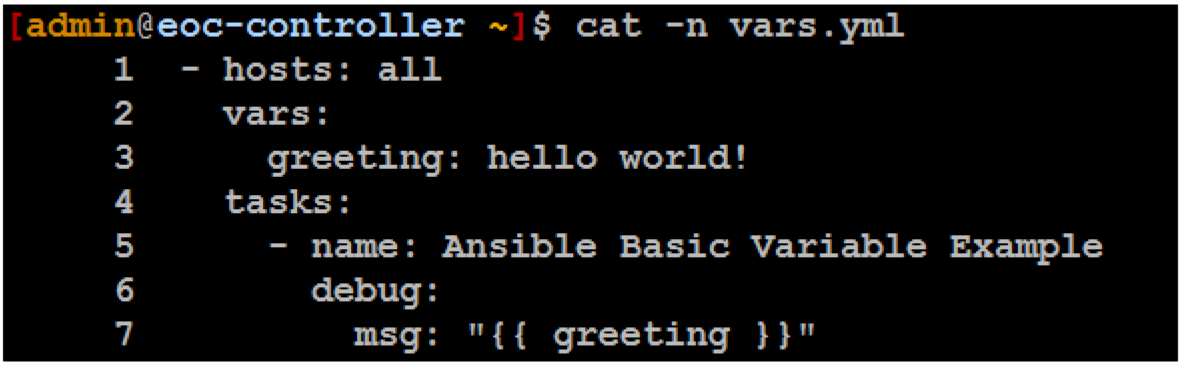


```
- hosts: all
  vars:
    greeting: hello world!
  tasks:
    - name: Ansible Basic Variable Example
      debug:
        msg: "{{ greeting }}"
```

**1.3** Let’s view the **vars.yml** file.

```
# cat -n vars.yml
```

**Output:**

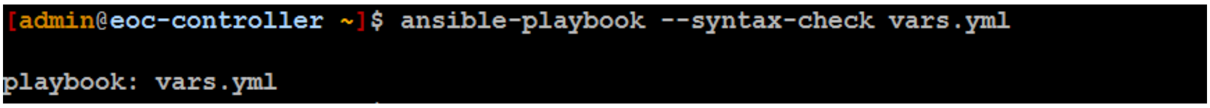


```
[admin@eoc-controller ~]$ cat -n vars.yml
     1  - hosts: all
     2    vars:
     3      greeting: hello world!
     4    tasks:
     5      - name: Ansible Basic Variable Example
     6        debug:
     7          msg: "{{ greeting }}"
```

**1.4** Let’s verify the syntax of **vars.yml** manifest by executing below command

```
# ansible-playbook --syntax-check vars.yml
```

**Output:**

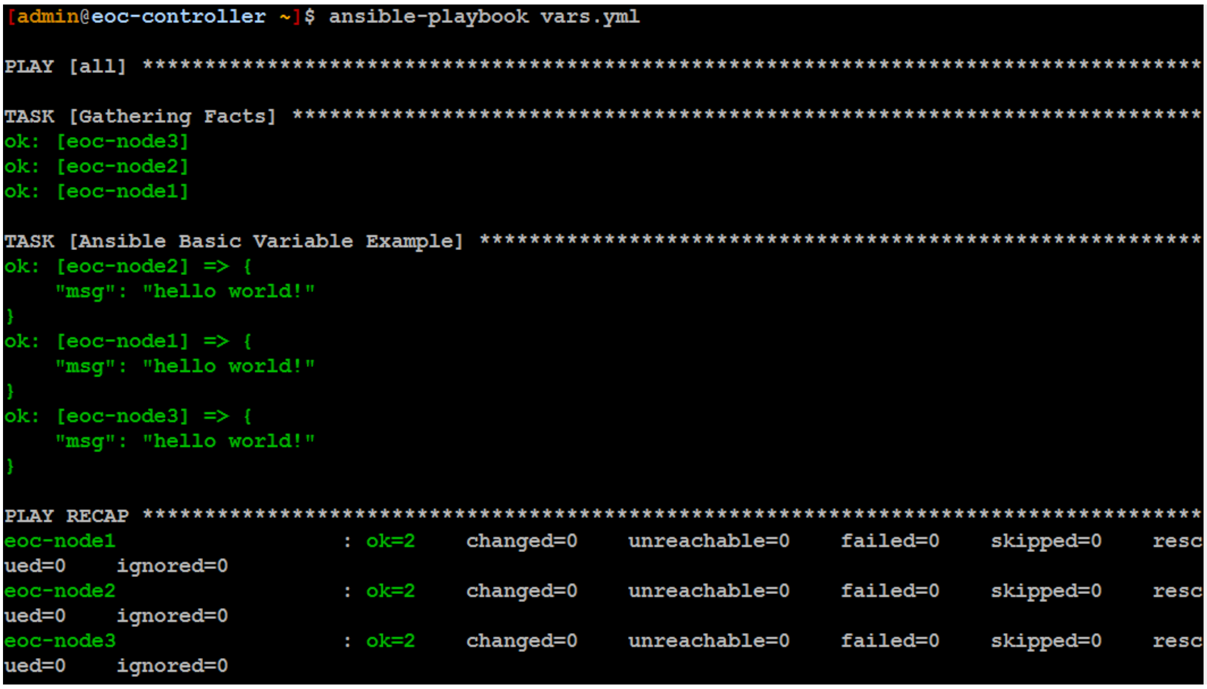


```
[admin@eoc-controller ~]$ ansible-playbook --syntax-check vars.yml

playbook: vars.yml
```

**1.5** Let's deploy the manifest file against all host.

```
# ansible-playbook vars.yml
```

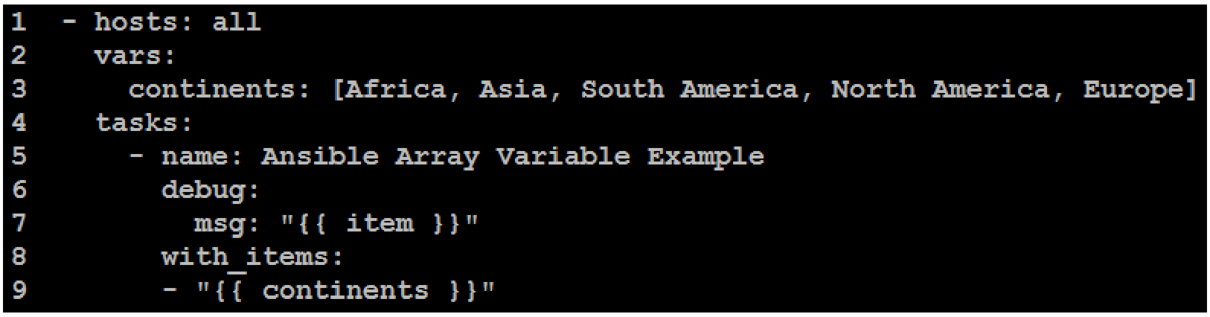
**Output:**



```
[admin@eoc-controller ~]$ ansible-playbook vars.yml

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

TASK [Gathering Facts] *************************************************************
ok: [eoc-node3]
ok: [eoc-node2]
ok: [eoc-node1]

TASK [Ansible Basic Variable Example] **********************************************
ok: [eoc-node2] => {
    "msg": "hello world!"
}
ok: [eoc-node1] => {
    "msg": "hello world!"
}
ok: [eoc-node3] => {
    "msg": "hello world!"
}

PLAY RECAP *************************************************************************
eoc-node1                  : ok=2    changed=0    unreachable=0    failed=0    skipped=0    resc
ued=0    ignored=0
eoc-node2                  : ok=2    changed=0    unreachable=0    failed=0    skipped=0    resc
ued=0    ignored=0
eoc-node3                  : ok=2    changed=0    unreachable=0    failed=0    skipped=0    resc
ued=0    ignored=0
```

Additionally, you can have a list or an array of variables as shown:

**1.6** The **vars2.yml** playbook below shows a variable called **continents**. The variable holds 5 different values – continent names.

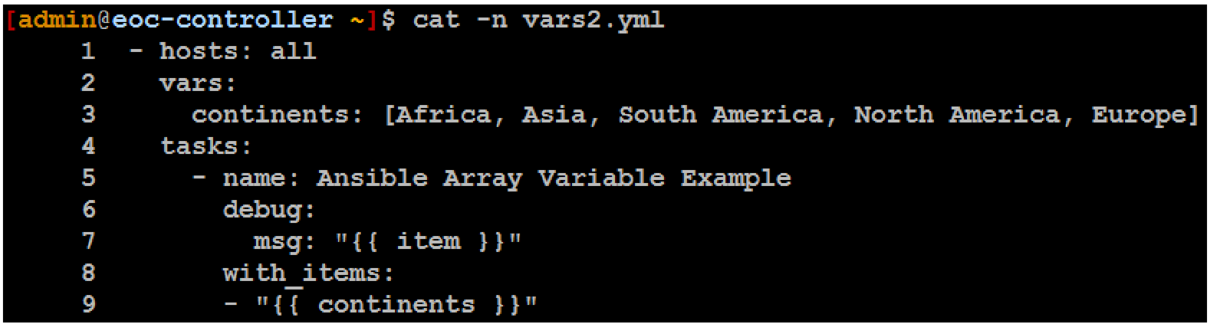


```
- hosts: all
  vars:
    continents: [Africa, Asia, South America, North America, Europe]
  tasks:
    - name: Ansible Array Variable Example
      debug:
        msg: "{{ item }}"
      with_items:
      - "{{ continents }}"
```

**1.7** Let's view the **vars2.yml** file.

```
# cat -n vars2.yml
```

**Output:**

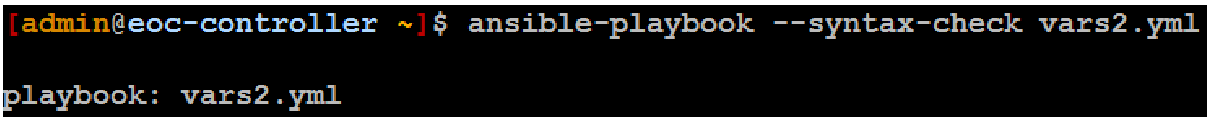


```
[admin@eoc-controller ~]$ cat -n vars2.yml
     1  - hosts: all
     2    vars:
     3      continents: [Africa, Asia, South America, North America, Europe]
     4    tasks:
     5      - name: Ansible Array Variable Example
     6        debug:
     7          msg: "{{ item }}"
     8        with_items:
     9        - "{{ continents }}"
```

**1.8** Let's verify the syntax of **vars2.yml** manifest by executing below command

```
# ansible-playbook --syntax-check vars2.yml
```

**Output:**

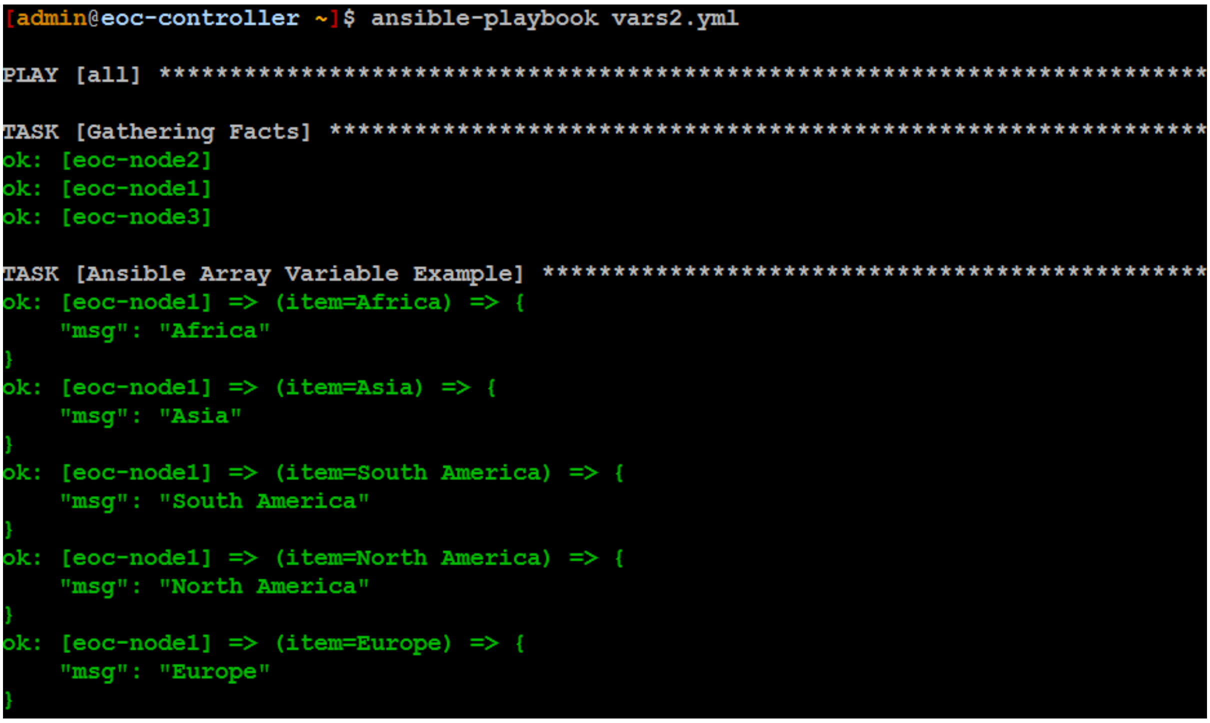


```
[admin@eoc-controller ~]$ ansible-playbook --syntax-check vars2.yml

playbook: vars2.yml
```

**1.9** Let's deploy the **vars2.yml** manifest on all the host.

```
# ansible-playbook vars2.yml
```

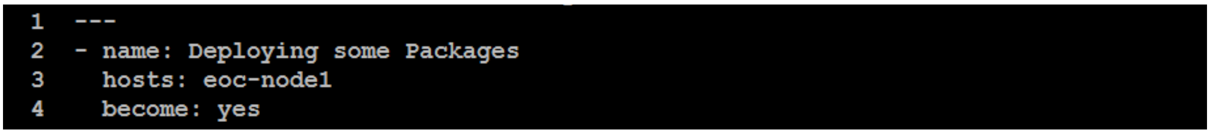
**Output:**



```
[admin@eoc-controller ~]$ ansible-playbook vars2.yml

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

TASK [Gathering Facts] *********************************************************
ok: [eoc-node2]
ok: [eoc-node1]
ok: [eoc-node3]

TASK [Ansible Array Variable Example] ******************************************
ok: [eoc-node1] => (item=Africa) => {
    "msg": "Africa"
}
ok: [eoc-node1] => (item=Asia) => {
    "msg": "Asia"
}
ok: [eoc-node1] => (item=South America) => {
    "msg": "South America"
}
ok: [eoc-node1] => (item=North America) => {
    "msg": "North America"
}
ok: [eoc-node1] => (item=Europe) => {
    "msg": "Europe"
}
```

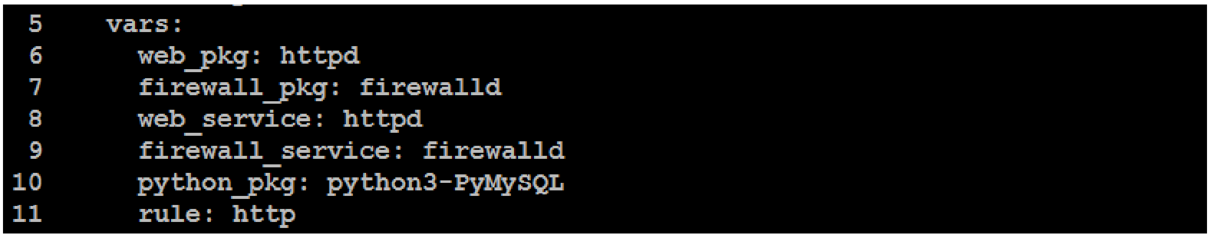
**Truncated….**

**1.10** Let's create a playbook **variable.yml** which install packages by calling variable.



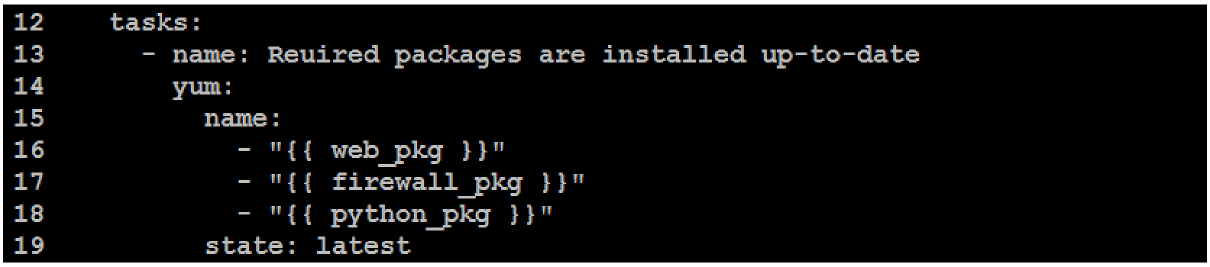
```
---
- name: Deploying some Packages
  hosts: eoc-node1
  become: yes
```

**1.11** Let’s define the following variables in vars section.



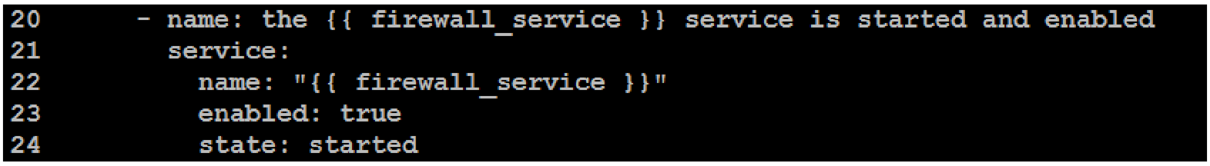
```
  vars:
    web_pkg: httpd
    firewall_pkg: firewalld
    web_service: httpd
    firewall_service: firewalld
    python_pkg: python3-PyMySQL
    rule: http
```

**1.12** Let’s create the task which should use the yum module to make sure the latest versions of the required packages are installed.



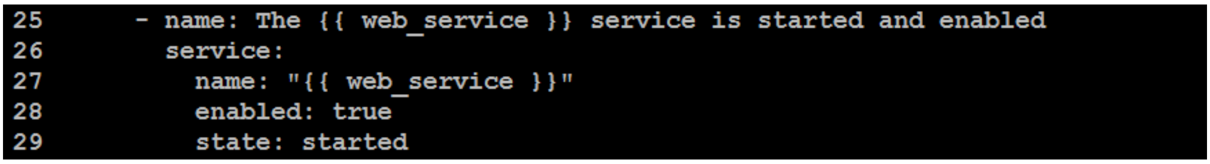
```
  tasks:
    - name: Reuired packages are installed up-to-date
      yum:
        name:
          - "{{ web_pkg }}"
          - "{{ firewall_pkg }}"
          - "{{ python_pkg }}"
        state: latest
```

**1.13** Let’s create the task which enable the firewall service.



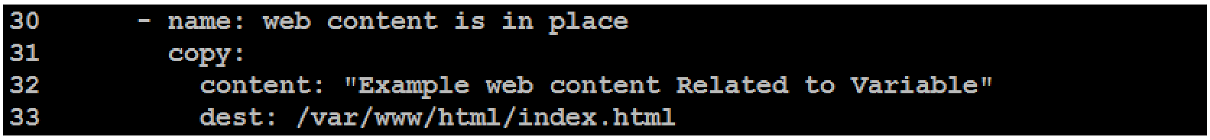
```
    - name: the {{ firewall_service }} service is started and enabled
      service:
        name: "{{ firewall_service }}"
        enabled: true
        state: started
```

**1.14** Let’s create the task which enable the httpd service svc.



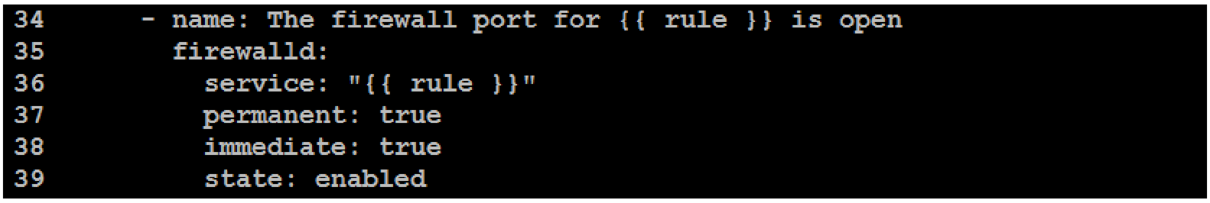
```
    - name: The {{ web_service }} service is started and enabled
      service:
        name: "{{ web_service }}"
        enabled: true
        state: started
```

**1.15** Add a task that ensures specific content exists in the /var/www/html/index.html file.



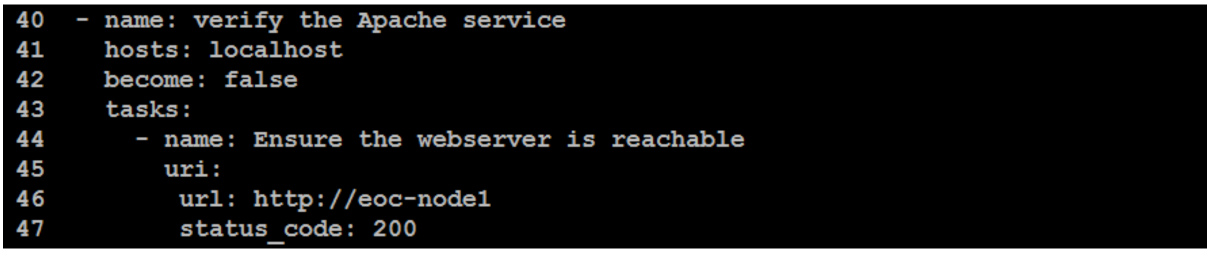
```
    - name: web content is in place
      copy:
        content: "Example web content Related to Variable"
        dest: /var/www/html/index.html
```

**1.16** Add a task that uses the firewalld module to ensure the firewall ports are open for the firewalld service named in the rule variable.



```
    - name: The firewall port for {{ rule }} is open
      firewalld:
        service: "{{ rule }}"
        permanent: true
        immediate: true
        state: enabled
```

**1.17** Let’s create another play to verify the reach.

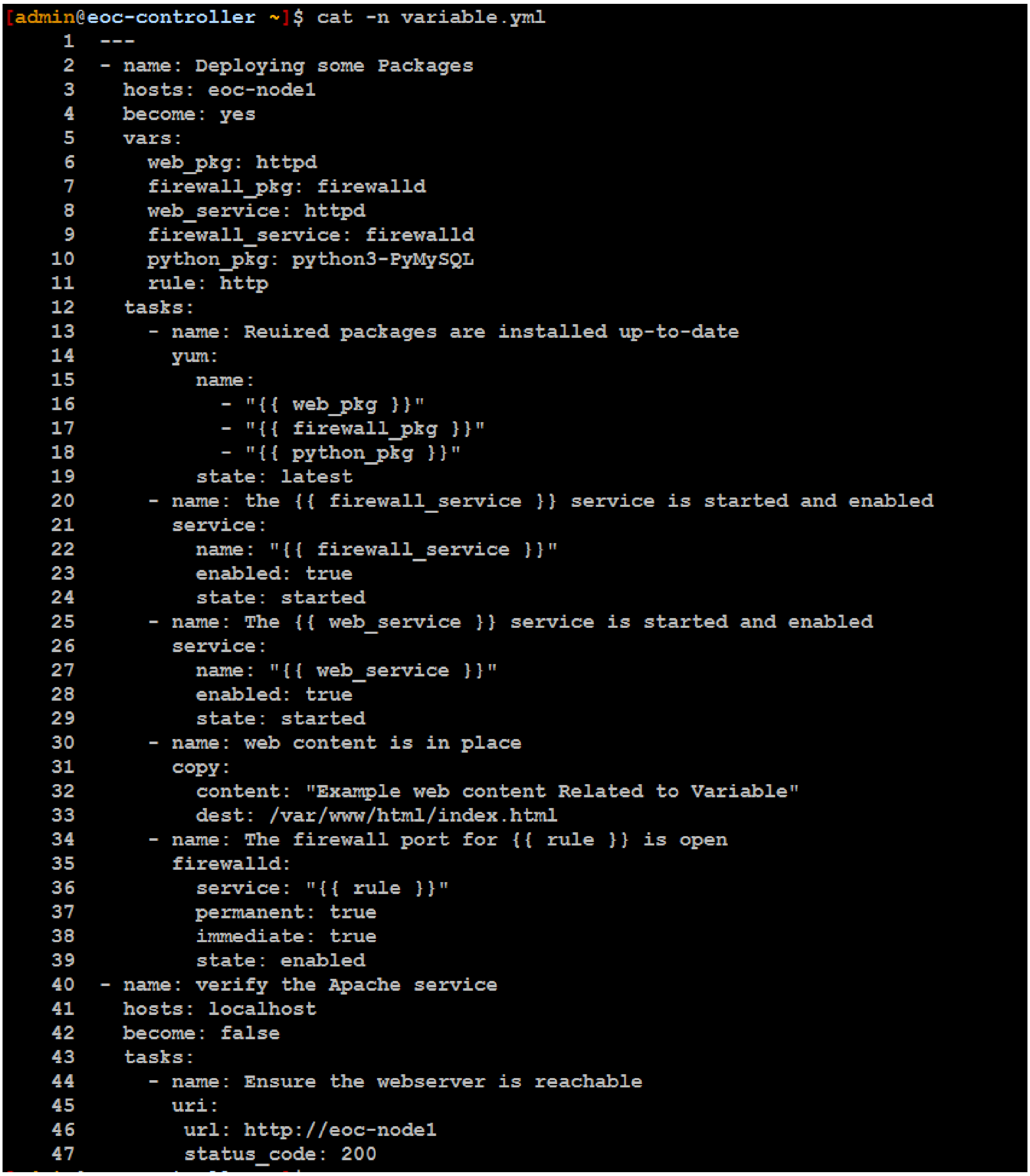


```
40   - name: verify the Apache service
41     hosts: localhost
42     become: false
43     tasks:
44       - name: Ensure the webserver is reachable
45         uri:
46          url: http://eoc-node1
47          status_code: 200
```

**1.18** Let’s view the **variable.yml** file.

```
# cat -n variable.yml
```

**Output:**

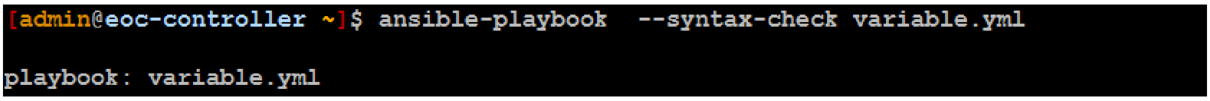


```
[admin@eoc-controller ~]$ cat -n variable.yml
     1  ---
     2  - name: Deploying some Packages
     3    hosts: eoc-node1
     4    become: yes
     5    vars:
     6      web_pkg: httpd
     7      firewall_pkg: firewalld
     8      web_service: httpd
     9      firewall_service: firewalld
    10      python_pkg: python3-PyMySQL
    11      rule: http
    12    tasks:
    13      - name: Reuired packages are installed up-to-date
    14        yum:
    15          name:
    16            - "{{ web_pkg }}"
    17            - "{{ firewall_pkg }}"
    18            - "{{ python_pkg }}"
    19          state: latest
    20      - name: the {{ firewall_service }} service is started and enabled
    21        service:
    22          name: "{{ firewall_service }}"
    23          enabled: true
    24          state: started
    25      - name: The {{ web_service }} service is started and enabled
    26        service:
    27          name: "{{ web_service }}"
    28          enabled: true
    29          state: started
    30      - name: web content is in place
    31        copy:
    32          content: "Example web content Related to Variable"
    33          dest: /var/www/html/index.html
    34      - name: The firewall port for {{ rule }} is open
    35        firewalld:
    36          service: "{{ rule }}"
    37          permanent: true
    38          immediate: true
    39          state: enabled
    40  - name: verify the Apache service
    41    hosts: localhost
    42    become: false
    43    tasks:
    44      - name: Ensure the webserver is reachable
    45        uri:
    46         url: http://eoc-node1
    47         status_code: 200
```

**1.19** Let's verify the syntax of **variable.yml** by executing below command

```
# ansible-playbook --syntax-check variable.yml
```

**Output:**

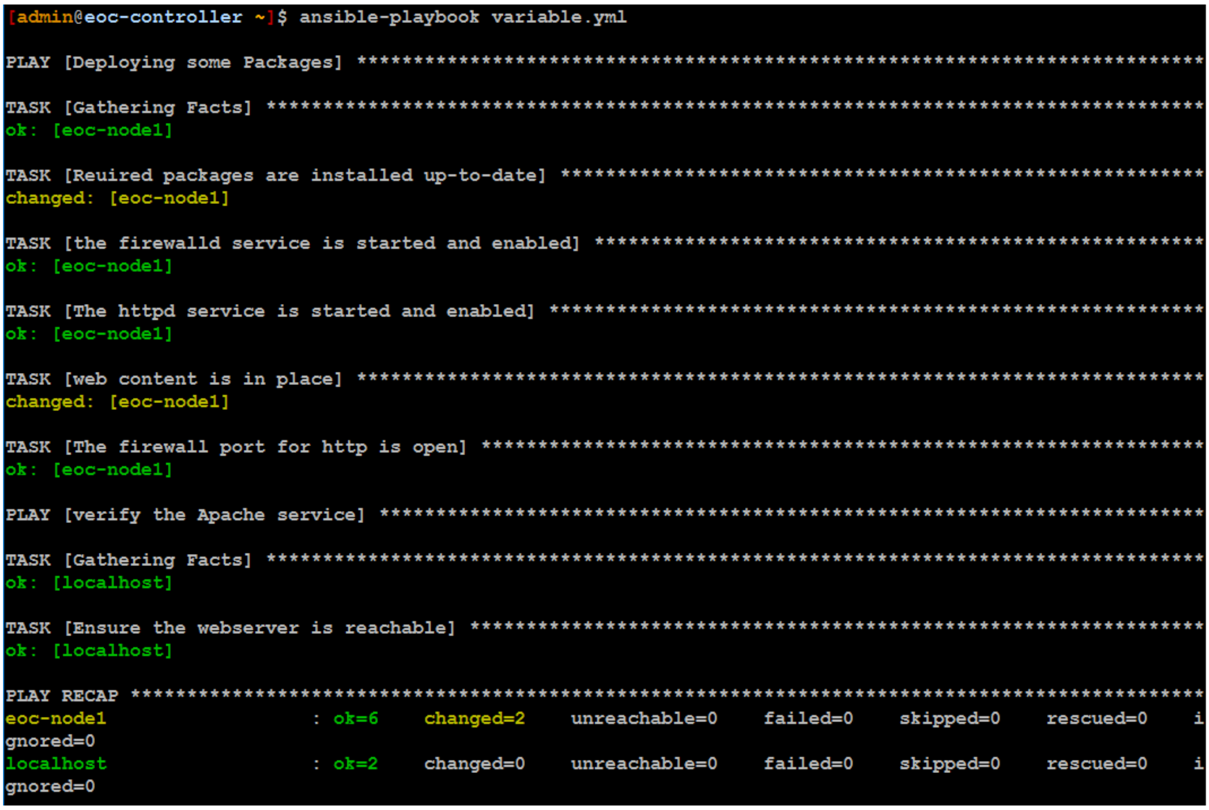


```
[admin@eoc-controller ~]$ ansible-playbook  --syntax-check variable.yml

playbook: variable.yml
```

**1.20** Let's run the ansible playbook command to run the manifest **variable.yml**

```
# ansible-playbook variable.yml
```

**Output:**

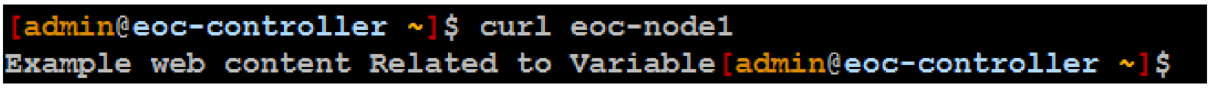


```
[admin@eoc-controller ~]$ ansible-playbook variable.yml

PLAY [Deploying some Packages] ***************************************************************

TASK [Gathering Facts] ***********************************************************************
ok: [eoc-node1]

TASK [Reuired packages are installed up-to-date] *********************************************
changed: [eoc-node1]

TASK [the firewalld service is started and enabled] ******************************************
ok: [eoc-node1]

TASK [The httpd service is started and enabled] **********************************************
ok: [eoc-node1]

TASK [web content is in place] ***************************************************************
changed: [eoc-node1]

TASK [The firewall port for http is open] ****************************************************
ok: [eoc-node1]

PLAY [verify the Apache service] *************************************************************

TASK [Gathering Facts] ***********************************************************************
ok: [localhost]

TASK [Ensure the webserver is reachable] *****************************************************
ok: [localhost]

PLAY RECAP ***********************************************************************************
eoc-node1                  : ok=6    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    i
gnored=0
localhost                  : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    i
gnored=0
```

**1.21** Use the curl command to verify that **eoc-node1** is configured as an HTTPD Server

```
# curl eoc-node1
```

**Output:**



```
[admin@eoc-controller ~]$ curl eoc-node1
Example web content Related to Variable[admin@eoc-controller ~]$
```