

Seif Eldin Mohamed Aboelhassan Mohamed Hendawy

Lab 3

1. Loop over a list of packages and install latest versions.

```
- name: Lab 3 Plays
hosts: all
tasks:
  - name: Install the latest version of packages
    apt:
      name: "{{ item }}"
      state: latest
    loop:
      - nginx
      - mysql-server
      - postgresql
      - apache2
      - git
```

```
seif@seif-Alienware-17-R5:~/Documents/Ansible/Day01$ ansible-playbook Lab3.yml
BECOME password:

PLAY [Lab 3 Plays] *****

TASK [Gathering Facts] *****
ok: [172.17.0.2]

TASK [Install the latest version of packages] *****
ok: [172.17.0.2] => (item=nginx)
changed: [172.17.0.2] => (item=mysql-server)
changed: [172.17.0.2] => (item=postgresql)
changed: [172.17.0.2] => (item=apache2)
changed: [172.17.0.2] => (item=git)

PLAY RECAP *****
172.17.0.2 : ok=2 changed=1 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
```

2. Loop over a list of packages and use different states as per input.

```
Lab3.yml vars_file x
Day01 > vars > vars_file > Google Cloud Cod
1 ---
2 packages:
3   - name: nginx
4     state: latest
5   - name: mysql-server
6     state: present
7   - name: postgresql
8     state: absent
9   - name: apache2
10    state: present
11   - name: git
12    state: absent
```

```

Lab3.yml x vars_file x
ay01 > A Lab3.yml
1 - name: Lab 3 Plays
2   hosts: all
3   vars_files:
4     - vars/vars_file
5   tasks:
6     - name: Install packages eith Different States
7       apt:
8         name: "{{ item.name }}"
9         state: "{{ item.state }}"
10        loop: "{{ packages }}"

```

```

● seif@seif-Alienware-17-R5:~/Documents/Ansible/Day01$ ansible-playbook Lab3.yml
BECOME password:

PLAY [Lab 3 Plays] *****

TASK [Gathering Facts] *****
ok: [172.17.0.2]

TASK [Install packages eith Different States] *****
ok: [172.17.0.2] => (item={'name': 'nginx', 'state': 'latest'})
ok: [172.17.0.2] => (item={'name': 'mysql-server', 'state': 'present'})
changed: [172.17.0.2] => (item={'name': 'postgresql', 'state': 'absent'})
ok: [172.17.0.2] => (item={'name': 'apache2', 'state': 'present'})
changed: [172.17.0.2] => (item={'name': 'git', 'state': 'absent'})

PLAY RECAP *****
172.17.0.2      : ok=2    changed=1    unreachable=0    failed=0    skipped=0

○ seif@seif-Alienware-17-R5:~/Documents/Ansible/Day01$ 

```

3. Install nginx or httpd depending on distribution
4. Restart nginx service if distribution is ubuntu.
5. Restart httpd service if distribution is centos.

```
- name: Install nginx or httpd depending on distribution
  apt:
    name: "{{ 'nginx' if ansible_distribution == 'Ubuntu' else 'httpd' }}"
    state: latest
    when: "'Ubuntu' in ansible_distribution or 'CentOS' in ansible_distribution"

- name: Restart nginx service if distribution is ubuntu
  sysvinit:
    name: nginx
    state: restarted
    when: "'Ubuntu' in ansible_distribution"

- name: Restart httpd service if distribution is centos
  sysvinit:
    name: httpd
    state: restarted
    when: "'CentOS' in ansible_distribution"
```

```
● seif@seif-Alienware-17-R5:~/Documents/Ansible/Day01$ ansible-playbook Lab3.yml
BECOME password:

PLAY [Lab 3 Plays] *****

TASK [Gathering Facts] *****
ok: [172.17.0.2]

TASK [Install nginx or httpd depending on distribution] *****
ok: [172.17.0.2]

TASK [Restart nginx service if distribution is ubuntu] *****
changed: [172.17.0.2]

TASK [Restart httpd service if distribution is centos] *****
skipping: [172.17.0.2]

PLAY RECAP *****
172.17.0.2 : ok=3 changed=1 unreachable=0 failed=0 skipped=1 rescued=0 ignored=0

○ seif@seif-Alienware-17-R5:~/Documents/Ansible/Day01$
```

6. View the value of your register variable using debug module
7. Restart service if the installation task was changed or was not failed

```
- name: Install nginx or httpd depending on distribution
  apt:
    name: "{{ 'nginx' if ansible_distribution == 'Ubuntu' else 'httpd' }}"
    state: latest
  when: "'Ubuntu' in ansible_distribution or 'CentOS' in ansible_distribution"
  register: res

- name: View installation result
  debug:
    var: res

- name: Restart nginx service if distribution is ubuntu
  sysvinit:
    name: nginx
    state: restarted
  when:
    - "'Ubuntu' in ansible_distribution"
    - res.changed == false or res.failed == false

- name: Restart httpd service if distribution is centos
  sysvinit:
    name: httpd
    state: restarted
  when:
    - "'CentOS' in ansible_distribution"
    - res.changed == false or res.failed == false
```

```
seif@seif-Alienware-17-R5:~/Documents/Ansible/Day01$ ansible-playbook Lab3.yml
BECOME password:

PLAY [Lab 3 Plays] *****

TASK [Gathering Facts] *****
ok: [172.17.0.2]

TASK [Install nginx or httpd depending on distribution] *****
ok: [172.17.0.2]

TASK [View installation result] *****
ok: [172.17.0.2] => {
  "res": {
    "cache_update_time": 1694765850,
    "cache_updated": false,
    "changed": false,
    "failed": false
  }
}

TASK [Restart nginx service if distribution is ubuntu] *****
changed: [172.17.0.2]

TASK [Restart httpd service if distribution is centos] *****
skipping: [172.17.0.2]

PLAY RECAP *****
172.17.0.2          : ok=4    changed=1    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

8. Install a package (get the package name from vars)
9. copy a list of files (index.html & file.css & file.js) from controller to host using loop (get the list of file names from vars) (the actual files will be stored in ./your_current_directory)(will be defined as Handler)(will be triggered by step 1)
10. Restart the service of the installed package (will be defined as Handler) (will be triggered by step 2)

```
- name: Lab 3 Handlers Plays
hosts: all
vars:
  - pkg_name: nginx
tasks:
  - name: Install Package
    apt:
      name: "{{ pkg_name }}"
      state: present

  - name: Copy files from controller to host
    copy:
      src: "{{ item }}"
      dest: /var/www/html/{{ item | basename }}
    loop:
      - "/home/seif/Documents/Ansible/Day01/index.html"
      - "/home/seif/Documents/Ansible/Day01/file.css"
      - "/home/seif/Documents/Ansible/Day01/file.js"
    notify: Restart service

handlers:
  - name: Restart service
    sysvinit:
      name: "{{ pkg_name }}"
      state: restarted
```

```
seif@seif-Alienware-17-R5:~/Documents/Ansible/Day01$ ansible-playbook Handlers.yml
BECOME password:

PLAY [Lab 3 Handlers Plays] *****

TASK [Gathering Facts] *****
ok: [172.17.0.2]

TASK [Install Package] *****
ok: [172.17.0.2]

TASK [Copy files from controller to host] *****
ok: [172.17.0.2] => (item=/home/seif/Documents/Ansible/Day01/index.html)
changed: [172.17.0.2] => (item=/home/seif/Documents/Ansible/Day01/file.css)
changed: [172.17.0.2] => (item=/home/seif/Documents/Ansible/Day01/file.js)

RUNNING HANDLER [Restart service] *****
changed: [172.17.0.2]

PLAY RECAP *****
172.17.0.2          : ok=4    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

seif@seif-Alienware-17-R5:~/Documents/Ansible/Day01$
```

11. installing a package (get the package name from vars)
12. Copying index.html from controller to host using template (get the template name & template message from vars)(the actual template file will be stored in ./your_current_directory) (will trigger step 4)
13. copying a list of files (file.css & file.js) from controller to host using loop (get the list of file names from vars)(the actual files will be stored in ./your_current_directory)(will be defined as Handler)(will be triggered by step 1)
14. Restart the service of the installed package (will be defined as Handler)(will be triggered by step 3)

```
- name: Lab 3 Temps Plays
hosts: all
vars:
  - pkg_name: nginx
  - tmp_name: index.html.j2
  - tmp_message: "Hello, My Name is Seif Hendawy, I'm A DevOps Engineer.<br> This is the Ansible course From ITI"
tasks:
  - name: Install Package
    apt:
      name: "{{ pkg_name }}"
      state: present

  - name: Copy index from controller to host
    template:
      src: "/home/seif/Documents/Ansible/Day01/{{ tmp_name }}"
      dest: "/var/www/html/index.html"
    notify: Copy files from controller to host

  - name: Copy css & js files from controller to host
    copy:
      src: "/home/seif/Documents/Ansible/Day01/{{ item }}"
      dest: "/var/www/html/{{ item }}"
    loop:
      - "file.css"
      - "file.js"
    notify: Restart service

handlers:
  - name: Copy files from controller to host
    debug:
      msg: "Copying files from controller to host"
  - name: Restart service
    sysvinit:
      name: "{{ pkg_name }}"
      state: restarted
```

```

seif@seif-Alienware-17-R5:~/Documents/Ansible/Day01$ ansible-playbook Temps.yml
BECOME password:

PLAY [Lab 3 Temps Plays] *****

TASK [Gathering Facts] *****
ok: [172.17.0.2]

TASK [Install Package] *****
ok: [172.17.0.2]

TASK [Copy index from controller to host] *****
changed: [172.17.0.2]

TASK [Copy css & js files from controller to host] *****
changed: [172.17.0.2] => (item=file.css)
ok: [172.17.0.2] => (item=file.js)

RUNNING HANDLER [Copy files from controller to host] *****
ok: [172.17.0.2] => {
  "msg": "Copying files from controller to host"
}

RUNNING HANDLER [Restart service] *****
changed: [172.17.0.2]

PLAY RECAP *****
172.17.0.2          : ok=6    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

```

⚠ Not secure | 172.17.0.2

**Hello, My Name is Seif Hendawy, I'm A DevOps Engineer.
This is the Ansible course From ITI**