



najah.png

Al-Najah National University
Computer Network and Information Security
Network Administration Lab

Automation 2

Instructor:Dr. Ahmed Awad

Dima Bshara
Mohammed Adnan

3. After that, we add the task **update_cache: yes** in ansible that equal apt update as figure 3 shown.

```
student@Ubuntu-desktop:~/nislabs$ cat install_apache.yml
---

- hosts: all
  become: true
  tasks:
    - name: update repository index
      apt:
        update_cache: yes
    - name: install apache2 package
      apt:
        name: apache2
```

Figure 3: PlayBook File afte add update.

4. Then we run the playbook and notice the tasks that are executed successfully. **ok=3** which are the gathering fact,update repository, and install apache2 package tasks as figure 4 shown.

```
student@Ubuntu-desktop:~/nislabs$ ansible-playbook --ask-become-pass install_apache.yml
BECOME password:

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [172.16.107.53]
ok: [172.16.107.81]
ok: [172.16.107.39]

TASK [update repository index] *****
changed: [172.16.107.39]
changed: [172.16.107.53]
changed: [172.16.107.81]

TASK [install apache2 package] *****
ok: [172.16.107.81]
ok: [172.16.107.39]
ok: [172.16.107.53]

PLAY RECAP *****
172.16.107.39      : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
172.16.107.53      : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
172.16.107.81      : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

student@Ubuntu-desktop:~/nislabs$
```

Figure 4: Run yml file.

5. We added php support to the Apache server as figure 5 show then we run the playbook to make changes as figure 6 show. This playbook will install apache2 and libapache2mod-php packages if they are not installed but it won't update them if there are updates available. To make the playbook capable of updating packages we need to use the **state** parameter as figure 7
- state: latest will make sure the package is always the latest one available.

```
student@Ubuntu-desktop:~/nislabs$ cat install_apache.yml
```

```
---
- hosts: all
  become: true
  tasks:
    - name: update repository index
      apt:
        update_cache: yes
    - name: install apache2 package
      apt:
        name: apache2
    - name: add php support for apache
      apt:
        name: libapache2-mod-php
```

Figure 5: Add php support to playbook file.

```
TASK [add php support for apache] *****
changed: [172.16.107.81]
changed: [172.16.107.53]
changed: [172.16.107.39]

PLAY RECAP *****
172.16.107.39      : ok=4    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
172.16.107.53      : ok=4    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
172.16.107.81      : ok=4    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

student@Ubuntu-desktop:~/nislabs$
```

Figure 6: Run yml file.

```
---
- hosts: all
  become: true
  tasks:
    - name: update repository index
      apt:
        update_cache: yes
    - name: install apache2 package
      apt:
        name: apache2
        state: latest
    - name: add php support for apache
      apt:
        name: libapache2-mod-php
        state: latest
```

Figure 7: Add state to playbook file.

6. Also, we create another playbook that removes these packages the file called **remove_apache.yml** the **state: absent** parameter value means removing the package if present as figure 8 shown then run the playbook as figure 9 show.

```
- - -  
  
- hosts: all  
  become: true  
  tasks:  
  
    - name: remove apache2 package  
      apt:  
        name: apache2  
        state: absent  
  
    - name: remove php support for apache  
      apt:  
        name: libapache2-mod-php  
        state: absent
```

Figure 8: Remove package.

```
student@Ubuntu-desktop:~/nislalab$ ansible-playbook --ask-become-pass remove_apache.yml  
BECOME password:  
  
PLAY [all] *****  
  
TASK [Gathering Facts] *****  
ok: [172.16.107.81]  
ok: [172.16.107.39]  
ok: [172.16.107.53]  
  
TASK [remove apache2 package] *****  
changed: [172.16.107.81]  
changed: [172.16.107.53]  
changed: [172.16.107.39]  
  
TASK [remove php support for apache] *****  
changed: [172.16.107.81]  
changed: [172.16.107.39]  
changed: [172.16.107.53]  
  
PLAY RECAP *****  
172.16.107.39      : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0  
172.16.107.53      : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0  
172.16.107.81      : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0  
student@Ubuntu-desktop:~/nislalab$
```

Figure 9: Run yml file.

7. We try opening the site on one of our servers as figure 10 show can't open the apache server because the package was removed from all servers.
Then we run the install playbook file and return open the site it was accessed and opened because the package downloaded.

Unable to connect

Firefox can't establish a connection to the server at 172.16.107.81.

- The site could be temporarily unavailable or too busy. Try again in a few moments.
- If you are unable to load any pages, check your computer's network connection.
- If your computer or network is protected by a firewall or proxy, make sure that Firefox is permitted to access the Web.

Try Again

Figure 10: Try to open apache server.

8. We added two files to our nislalab directory which is connected to a git repository so we add both these files to github as figure 11 show .

```
student@Ubuntu-desktop:~/nislalab$ git add .
student@Ubuntu-desktop:~/nislalab$ git commit -m "install/remove by student1"
[main e8ada5d] install/remove by student1
2 files changed, 31 insertions(+)
 create mode 100644 install_apache.yml
 create mode 100644 remove_apache.yml
student@Ubuntu-desktop:~/nislalab$ git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 2 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 586 bytes | 293.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), done.
To github.com:mohammedix88/nislalab.git
 32b30d9..e8ada5d  main -> main
student@Ubuntu-desktop:~/nislalab$
```

Figure 11: Push the fils to github.

9. Then we pull it in workstation2 as figure 12 show .

```
172.16.107.81
student@Ubuntu-desktop:~/nislalab$ git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 4 (delta 1), reused 4 (delta 1), pack-reused 0
Unpacking objects: 100% (4/4), 566 bytes | 566.00 KiB/s, done.
From github.com:mohammedix88/nislalab
 32b30d9..e8ada5d  main      -> origin/main
Updating 32b30d9..e8ada5d
Fast-forward
 install_apache.yml | 16 +++++
 remove_apache.yml  | 15 +++++
2 files changed, 31 insertions(+)
 create mode 100644 install_apache.yml
 create mode 100644 remove_apache.yml
student@Ubuntu-desktop:~/nislalab$ ls
ansible.cfg  install_apache.yml  inventory  README.md  remove_apache.yml
student@Ubuntu-desktop:~/nislalab$
```

Figure 12: Push the fils to github.

1.2 The 'when' Conditional

The playbook we created will work fine if all servers are Debian-based systems; because we used the apt module. if some of the servers have a base other than Debian then the playbook will fail when used on them.

- (a) We modify the inventory file by adding the AlmaLinux server IP address. Then, running the playbook file that is just with module **apt** and noticed that the output for the AlmaLinux server was failed as figure 13 show.

```
PLAY RECAP *****
172.16.107.39      : ok=4    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
172.16.107.53      : ok=4    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
172.16.107.62      : ok=1    changed=0    unreachable=0    failed=1    skipped=0    rescued=0    ignored=0
172.16.107.81      : ok=4    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
student@ubuntu-desktop:~/nislabs$
```

Figure 13: Failed to install in AlmaLinux.

- (b) Then we gather information to know the distributions of our servers by running the gather_facts module as figure 14 show that AlmaLinux is a RedHat distribution.

```
student@Ubuntu-desktop:~/nislabs$ ansible all -m gather_facts | grep ansible_distribution
[WARNING]: Platform linux on host 172.16.107.62 is using the discovered Python interpreter
another Python interpreter could change this. See https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more
information.
  "ansible_distribution": "AlmaLinux",
  "ansible_distribution_file_parsed": true,
  "ansible_distribution_file_path": "/etc/redhat-release",
  "ansible_distribution_file_variety": "RedHat",
  "ansible_distribution_major_version": "8",
  "ansible_distribution_release": "Electric Cheetah",
  "ansible_distribution_version": "8.4",
  "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_distribution": "AlmaLinux",
  "ansible_distribution_file_parsed": true,
  "ansible_distribution_file_path": "/etc/redhat-release",
  "ansible_distribution_file_variety": "RedHat",
  "ansible_distribution_major_version": "8",
  "ansible_distribution_release": "Electric Cheetah",
  "ansible_distribution_version": "8.4",
  "ansible_distribution": "AlmaLinux",
  "ansible_distribution_file_parsed": true,
  "ansible_distribution_file_path": "/etc/redhat-release",
  "ansible_distribution_file_variety": "RedHat",
  "ansible_distribution_major_version": "8",
  "ansible_distribution_release": "Electric Cheetah",
  "ansible_distribution_version": "8.4"
```

Figure 14: Gather information.

- (c) So we modify the playbook that uses when condition to suitable for AlmaLinux then run the file and noticed that no failed as figure 15 show.

```
- name: update repository index
  dnf:
    update_cache: yes
  when: ansible_distribution == "AlmaLinux"

- name: install httpd package
  dnf:
    name: httpd
    state: latest
  when: ansible_distribution == "AlmaLinux"

- name: add php support for apache
  dnf:
    name: php
    state: latest
  when: ansible_distribution == "AlmaLinux"
student@Ubuntu-desktop:~/nislabs$
*****
172.16.107.39      : ok=4    changed=1    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
172.16.107.53      : ok=4    changed=1    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
172.16.107.62      : ok=4    changed=2    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
172.16.107.81      : ok=4    changed=1    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
student@Ubuntu-desktop:~/nislabs$
```

Figure 15: Playbook file.

2 Improving The Playbook

1. Our playbook includes many unnecessary lines that we can omit. So we did add multiple packages to the apt module to be installed on the system. This way, we need only one Task or play to install all needed packages. So we edit the playbook on workstation 2, as shown in figure 16, after that we run our play as figure 17 shown to make sure we have no syntax errors in the file.

```
---
- hosts: all
  become: true
  tasks:

    - name: update repository index
      apt:
        update_cache: yes
      when: ansible_distribution == "Ubuntu"

    - name: install apache2 and php packages for Ubuntu
      apt:
        name:
          - apache2
          - libapache2-mod-php
        state: latest
      when: ansible_distribution == "Ubuntu"

    - name: update repository index
      dnf:
        update_cache: yes
      when: ansible_distribution == "AlmaLinux"

    - name: install apache and php packages for AlmaLinux
      dnf:
        name:
          - httpd
          - php
        state: latest
      when: ansible_distribution == "AlmaLinux"
```

Figure 16: edit playbook file.

```
PLAY RECAP *****
172.16.107.39      : ok=3    changed=1    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
172.16.107.53      : ok=3    changed=1    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
172.16.107.62      : ok=3    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
172.16.107.81      : ok=3    changed=1    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
```

Figure 17: Run playbook file.

2. Since the update cache is a parameter of the apt module, we can also eliminate that task as shown in figure 18, now our playbook is down to 2 plays only, after that we run our playbook as figure 19 shown to make sure we have no syntax errors in the file.

```
---
- hosts: all
  become: true
  tasks:

    - name: install packages Ubuntu
      apt:
        name:
          - apache2
          - libapache2-mod-php
        state: latest
        update_cache: yes
        when: ansible_distribution == "Ubuntu"

    - name: install packages AlmaLinux
      dnf:
        name:
          - httpd
          - php
        state: latest
        update_cache: yes
        when: ansible_distribution == "AlmaLinux"
```

Figure 18: edit playbook file.

```
PLAY RECAP *****
172.16.107.39      : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
172.16.107.53      : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
172.16.107.62      : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
172.16.107.81      : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

Figure 19: Run playbook file.

3. Now we can even get our playbook down to one play using variables as shown in figures 20 and 21 package module is a generic package manager, which means it will use the default package manager of each distribution. the `{{apache_package}}` and `{{php_package}}` are variables we created and we can use any name we want. Now for this to work, we need to edit the inventory file to include the package names. Then we run our playbook as figure 22 shown to make sure we have no syntax errors in the file.

```
---
- hosts: all
  become: true
  tasks:
    - name: install apache and php
      package:
        name:
          - '{{apache_package}}'
          - '{{php_package}}'
        state: latest
        update_cache: yes
```

Figure 20: Run playbook file.

```
1 172.16.107.39 apache_package=apache2 php_package=libapache2-mod-php
2 172.16.107.53 apache_package=apache2 php_package=libapache2-mod-php
3 172.16.107.81 apache_package=apache2 php_package=libapache2-mod-php
4 172.16.107.62 apache_package=httpd php_package=php
```

Figure 21: Edit Inventory file.

```
PLAY RECAP *****
172.16.107.39      : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
172.16.107.53      : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
172.16.107.62      : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
172.16.107.81      : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

Figure 22: Run playbook file.

3 Targeting Specific Nodes

1. Now suppose that we have different roles that our servers have. Suppose we have web, database, and file servers. If we want to target the web servers; with certain tasks while doing other tasks for the file and web servers, then we need to categories our servers in the inventory file as shown in figure 23. After that, we edit the playbook file as figure 24 and add a host for every task and, the host will be the variable that we use in the inventory file to categories our servers. Then we run our playbook to made sure that only web server will install, as shown in figure 25.

```
1 [web_servers]
2 172.16.107.53
3 172.16.107.62
4 [db_servers]
5 172.16.107.39
6 172.16.107.62
7 [file_servers]
8 172.16.107.81
```

Figure 23: edit inventory file.

```
- hosts: web_servers
  become: true
  tasks:
    - name: install apache and php for Ubuntu servers
      apt:
        name:
          - apache2
          - libapache2-mod-php
        state: latest
        when: ansible_distribution == "Ubuntu"
    - name: install apache and php for AlmaLinux servers
      dnf:
        name:
          - httpd
          - php
        state: latest
        when: ansible_distribution == "AlmaLinux"
```

Figure 24: edit playbook file.

```

TASK [Gathering Facts] *****
ok: [172.16.107.62]
ok: [172.16.107.53]

TASK [install apache and php for Ubuntu servers] *****
skipping: [172.16.107.62]
ok: [172.16.107.53]

TASK [install apache and php for AlmaLinux servers] *****
skipping: [172.16.107.53]
ok: [172.16.107.62]

PLAY RECAP *****
172.16.107.39      : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
172.16.107.53      : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
172.16.107.62      : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
172.16.107.81      : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
student@Ubuntu-desktop:~/nislabs$

```

Figure 25: run playbook file.

2. After that, we add the database task as shown in figure 26, then we run the playbook to ensure that the webserver and database will install for a particular host as shown in figure 27. Also, we add a task for the fileserver host to install as shown in figure 28, then we run it again to make sure that all the packages install and there is no error as shown in figure 29.

```
- hosts: db_servers
  become: true
  tasks:

  - name: install Database package (AlmaLinux)
    dnf:
      name: mariadb
      state: latest
    when: ansible_distribution == "AlmaLinux"

  - name: install mariadb server
    apt:
      name: mariadb-server
      state: latest
    when: ansible_distribution == "Ubuntu"
```

Figure 26: edit playbook file & add database task.

```
PLAY RECAP *****
172.16.107.39      : ok=4    changed=1    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
172.16.107.53      : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
172.16.107.62      : ok=6    changed=1    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
172.16.107.81      : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
student@Ubuntu-desktop:~/nislab$
```

Figure 27: run playbook file.

```
-
5 - hosts: file_servers
7   become: true
3   tasks:
9
9     - name: install samba package
1       package:
2         name: samba
3         state: latest
```

Figure 28: edit playbook file & add file-server task.

```
TASK [install samba package] *****
changed: [172.16.107.81]

PLAY RECAP *****
172.16.107.39      : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
172.16.107.53      : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
172.16.107.62      : ok=6    changed=0    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
172.16.107.81      : ok=4    changed=1    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
student@Ubuntu-desktop:~/nislab$
```

Figure 29: run playbook file.

4 Using Tags

We can add tags to our tasks which is another way to execute plays on certain hosts. Suppose we want to run all plays on Ubuntu servers that have Apache installed. Tags are words we add to each task in the tag line.

1. So we add the tag under the task name as shown in figure 30, as we see that every task have tags, **tag: always** mean that it will always have done when any task run, then we list all available tagging in the playbook file by using list command as shown in figure 31. At first, we run our playbook using DB tag by using **ansible-playbook -tags DB -ask-become-pass site.yml** command, that command will install DB in every server have the DB tags as shown in figure 32. After that, we run our playbook, but now by using the alma tag by running **ansible-playbook -tags Alma -ask-become-pass site.yml** command, that command will run all tasks that contain the alma tag on it as shown in figure 33. In the end, we run the playbook to execute tasks, which will contain apache & DB tags, as the following command: **ansible-playbook -tags "apache, DB" -ask-become-pass site.yml**, figure 34 shows the result for that command.

```
when: ansible_distribution == "AlmaLinux"
- name: install updates (Ubuntu)
  tags: always
  apt:
    upgrade: dist
    update_cache: yes
  when: ansible_distribution == "Ubuntu"
- hosts: web_servers
  become: true
  tasks:
- name: install httpd package (AlmaLinux)
  tags: apache,Alma,httpd
  dnf:
    name:
      - httpd
      - php
    state: latest
  when: ansible_distribution == "AlmaLinux"
- name: install apache2 package (Ubuntu)
  tags: apache,apache2,ubuntu
  apt:
    name:
      - apache2
      - libapache2-mod-php
    state: latest
  when: ansible_distribution == "Ubuntu"
- hosts: db_servers
  become: true
  tasks:
- name: install mariadb server package (AlmaLinux)
  tags: Alma,db,mariadb
  dnf:
    name: mariadb
    state: latest
```

Figure 30: Edit playbook file by adding tags.

Figure 31: List all tags.

```

TASK [install mariadb server package (AlmaLinux)] *****
skipping: [172.16.107.39]
ok: [172.16.107.62]

TASK [install mariadb server] *****
skipping: [172.16.107.62]
ok: [172.16.107.39]

PLAY [file_servers] *****

PLAY RECAP *****
172.16.107.39      : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
172.16.107.53      : ok=3    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
172.16.107.62      : ok=5    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
172.16.107.81      : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
student@Ubuntu-desktop:~/nislabs$

```

Figure 32: Run playbook using db tags.

```

TASK [install httpd package (AlmaLinux)] *****
skipping: [172.16.107.53]
ok: [172.16.107.62]

PLAY [db_servers] *****

TASK [Gathering Facts] *****
ok: [172.16.107.62]
ok: [172.16.107.39]

TASK [install mariadb server package (AlmaLinux)] *****
skipping: [172.16.107.39]
ok: [172.16.107.62]

```

Figure 33: Run playbook using alma tags.

```

PLAY [web_servers] *****

TASK [Gathering Facts] *****
ok: [172.16.107.62]
ok: [172.16.107.53]

PLAY [db_servers] *****

TASK [Gathering Facts] *****
ok: [172.16.107.62]
ok: [172.16.107.39]

PLAY [file_servers] *****

PLAY RECAP *****

```

Figure 34: Run playbook using apache & db tags.