

Ansible – Lab2

Toka Farid Shawki

ITI- DevOps Track

Instructors: Mostafa Yehia & Ahmed Nabil

PLAYBOOK

1. Write your first playbook file.
2. Stop gather_facts and update cache.

```
! playbook.yml x  ansible.cfg
day1 > ! playbook.yml > Google Cloud Code > [ ] tasks > [ ] name
Ansible Playbook - Ansible playbook files (ansible.json)
1  - name: my_first_playbook
2    hosts: web_servers
3
4    tasks:
5      - name: health_check
6        ping:
7
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
● toka-farid@localhost:~/Ansible-labs/day1$ ansible-playbook playbook.yml
BECOME password:

PLAY [my_first_playbook] *****

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

TASK [health_check] *****
ok: [172.17.0.2]

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

○ toka-farid@localhost:~/Ansible-labs/day1$
```

```
! playbook.yml x  ! playbook_task2.yml ●  ansible.cfg
day1 > ! playbook.yml > [ ] tasks > { } apt > [ ] update_cache
1  - name: my_first_playbook
2    hosts: web_servers
3    gather_facts: false
4    tasks:
5
6      - name: Stop gather_facts and update cache
7        apt:
8          update_cache: true
9

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
● toka-farid@localhost:~/Ansible-labs/day1$ ansible-playbook playbook.yml
BECOME password:

PLAY [my_first_playbook] *****

TASK [Stop gather_facts and update cache] *****
changed: [172.17.0.2]

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

○ toka-farid@localhost:~/Ansible-labs/day1$
```

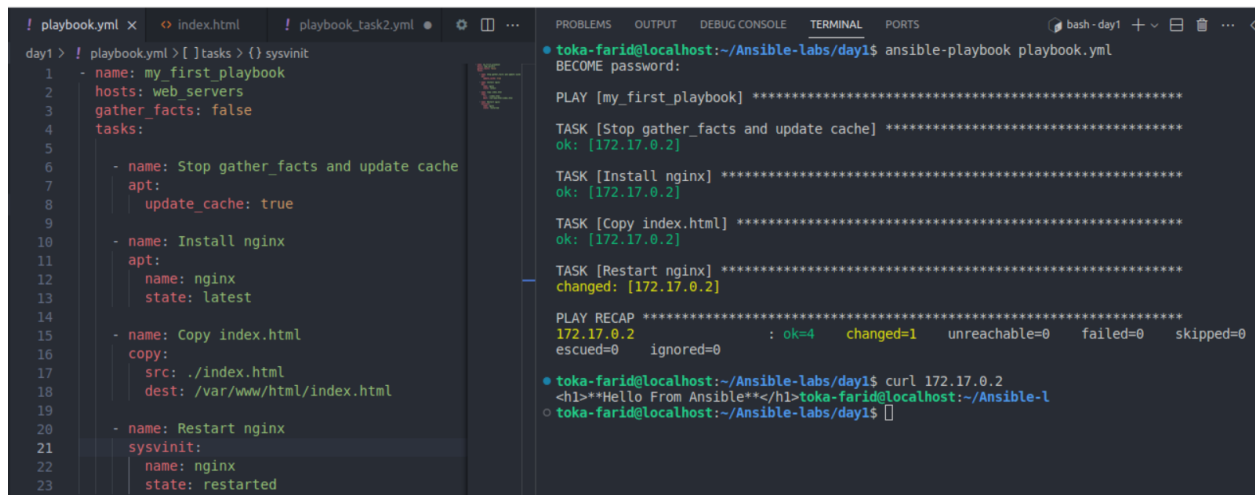
MODULES

Explore some built-in modules like:

(apt, dnf, package, service, command, copy, user, group, lineinfile, authorized_key, etc.)

ansible-builtin modules

1. Update cache.
2. Install the latest nginx.
3. Copy index.html from controller to host 1.
4. Restart nginx service.
5. Can you see your index.html file when you hit host 1 on port 80 ?



The screenshot shows an IDE with two main panes. The left pane displays the content of a file named `playbook.yml`. The right pane shows the terminal output of running the playbook.

playbook.yml content:

```
1 - name: my_first_playbook
2   hosts: web_servers
3   gather_facts: false
4   tasks:
5
6     - name: Stop gather_facts and update cache
7       apt:
8         update_cache: true
9
10    - name: Install nginx
11      apt:
12        name: nginx
13        state: latest
14
15    - name: Copy index.html
16      copy:
17        src: ./index.html
18        dest: /var/www/html/index.html
19
20    - name: Restart nginx
21      sysvinit:
22        name: nginx
23        state: restarted
```

Terminal Output:

```
● toka-farid@localhost:~/Ansible-labs/day1$ ansible-playbook playbook.yml
BECOME password:

PLAY [my_first_playbook] *****

TASK [Stop gather_facts and update cache] *****
ok: [172.17.0.2]

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

TASK [Copy index.html] *****
ok: [172.17.0.2]

TASK [Restart nginx] *****
changed: [172.17.0.2]

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

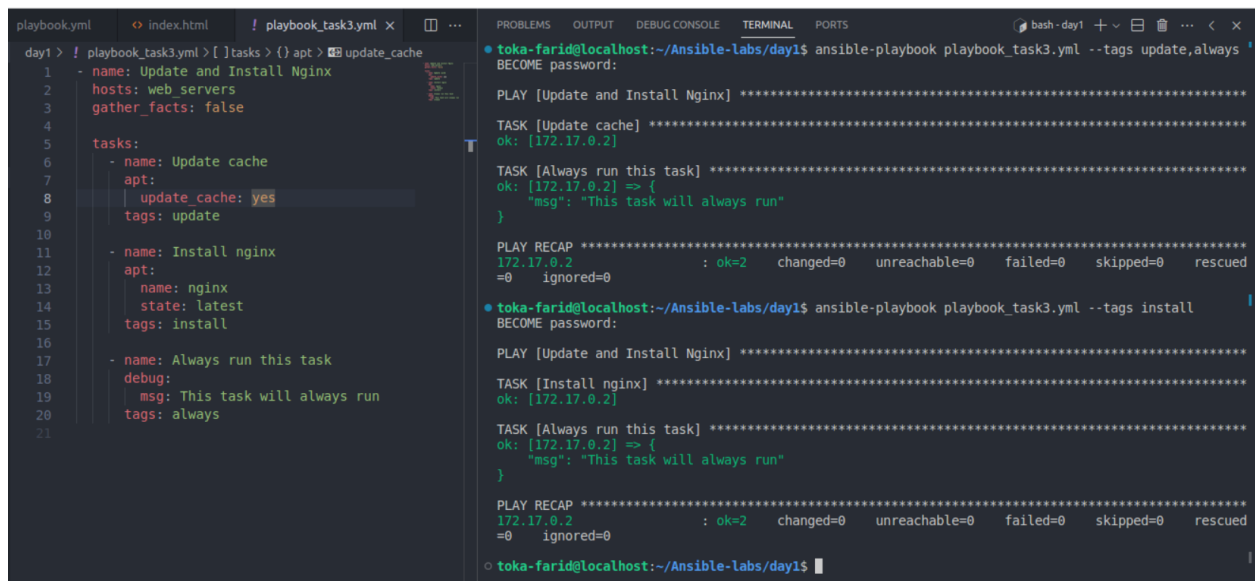
● toka-farid@localhost:~/Ansible-labs/day1$ curl 172.17.0.2
<h1>**Hello From Ansible**</h1>
● toka-farid@localhost:~/Ansible-l
```

TAGS

1. Write a simple playbook file.
2. Add two tasks (apt update – apt install nginx).
3. Add tags to the first task: update.
4. Add tags to the second task: install.
5. Run only the (apt update) task.

Example: ansible-playbook my-playbook.yml --tags update.

6. Add one task with “tags: always” and run the previous command again.



The screenshot shows an IDE with a file explorer on the left containing 'playbook.yml', 'index.html', and 'playbook_task3.yml'. The 'playbook_task3.yml' file is open and contains the following tasks:

```
1 - name: Update and Install Nginx
2   hosts: web_servers
3   gather_facts: false
4
5   tasks:
6     - name: Update cache
7       apt:
8         update_cache: yes
9         tags: update
10
11     - name: Install nginx
12       apt:
13         name: nginx
14         state: latest
15         tags: install
16
17     - name: Always run this task
18       debug:
19         msg: This task will always run
20       tags: always
21
```

The terminal on the right shows the execution of the playbook with the command `ansible-playbook playbook_task3.yml --tags update,always`. The output indicates that the 'Update cache' task was executed successfully, while the 'Install nginx' task was skipped because it does not have the 'update' tag.

```
● toka-farid@localhost:~/Ansible-labs/day1$ ansible-playbook playbook_task3.yml --tags update,always
BECOME password:

PLAY [Update and Install Nginx] *****
TASK [Update cache] *****
ok: [172.17.0.2]

TASK [Always run this task] *****
ok: [172.17.0.2] => {
  "msg": "This task will always run"
}

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

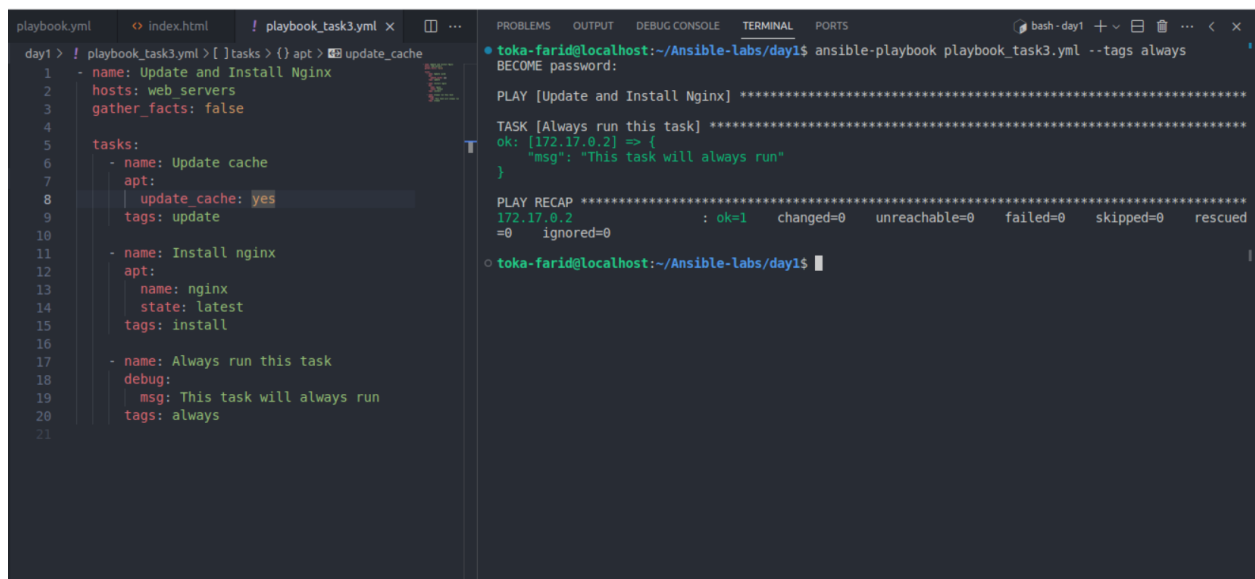
● toka-farid@localhost:~/Ansible-labs/day1$ ansible-playbook playbook_task3.yml --tags install
BECOME password:

PLAY [Update and Install Nginx] *****
TASK [Install nginx] *****
ok: [172.17.0.2]

TASK [Always run this task] *****
ok: [172.17.0.2] => {
  "msg": "This task will always run"
}

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

○ toka-farid@localhost:~/Ansible-labs/day1$
```



The screenshot shows the same IDE as before, but the terminal now shows the execution of the playbook with the command `ansible-playbook playbook_task3.yml --tags always`. The output indicates that the 'Always run this task' task was executed successfully, while the 'Update cache' and 'Install nginx' tasks were skipped because they do not have the 'always' tag.

```
● toka-farid@localhost:~/Ansible-labs/day1$ ansible-playbook playbook_task3.yml --tags always
BECOME password:

PLAY [Update and Install Nginx] *****
TASK [Always run this task] *****
ok: [172.17.0.2] => {
  "msg": "This task will always run"
}

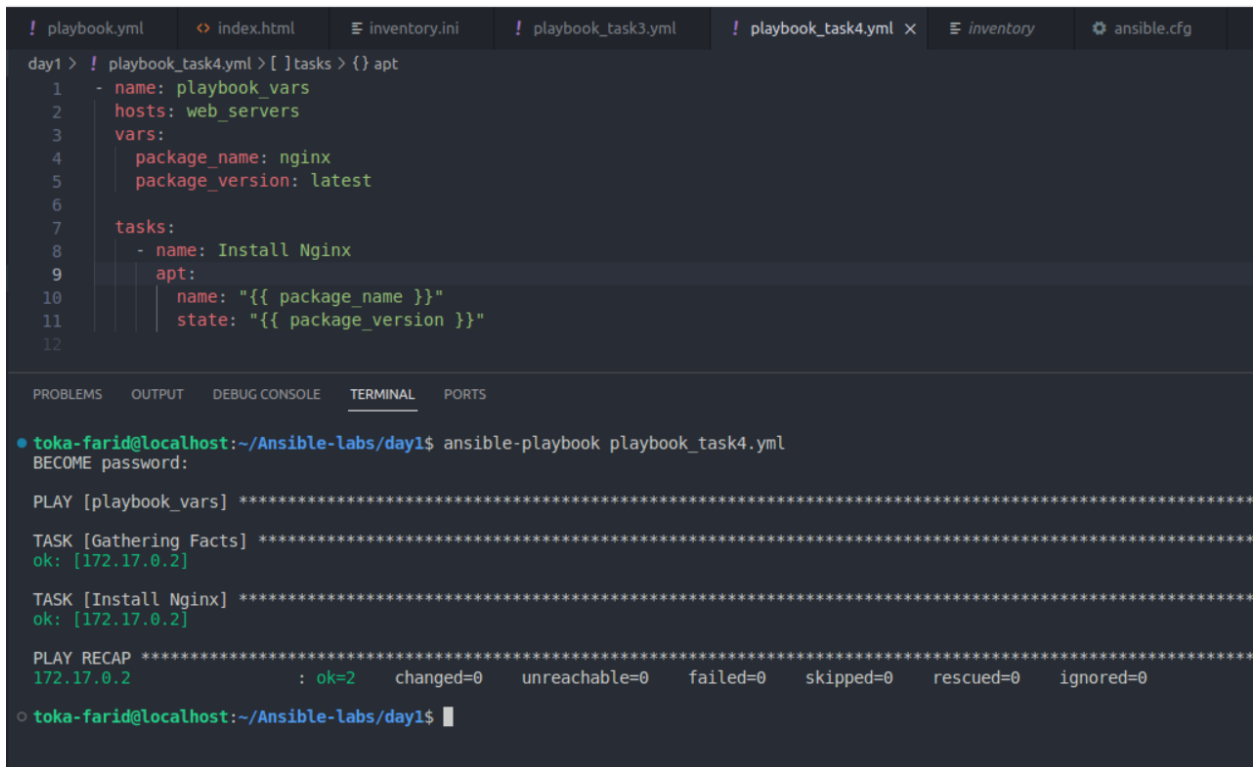
PLAY RECAP *****
172.17.0.2 : ok=1 changed=0 unreachable=0 failed=0 skipped=0 rescued
=0 ignored=0

○ toka-farid@localhost:~/Ansible-labs/day1$
```

VARIABLES

1. Define these variables (package_name, package_version).
 - on the playbook level.
 - on inventory level.
 - on command line level.
2. Use the apt module with the package name and version from your variables.

playbook level:



The screenshot shows an IDE with a dark theme. At the top, there are tabs for 'playbook.yml', 'index.html', 'inventory.ini', 'playbook_task3.yml', 'playbook_task4.yml', 'inventory', and 'ansible.cfg'. The 'playbook_task4.yml' tab is active, showing a YAML playbook. The playbook has a 'vars' section with 'package_name: nginx' and 'package_version: latest', and a 'tasks' section with a task named 'Install Nginx' using the 'apt' module with variables for name and state. Below the editor, there is a 'TERMINAL' tab showing the output of running 'ansible-playbook playbook_task4.yml'. The output shows the playbook running successfully on host 172.17.0.2, installing Nginx.

```
day1 > ! playbook_task4.yml > [ ] tasks > { } apt
1  - name: playbook_vars
2    hosts: web_servers
3    vars:
4      package_name: nginx
5      package_version: latest
6
7    tasks:
8      - name: Install Nginx
9        apt:
10          name: "{{ package_name }}"
11          state: "{{ package_version }}"
12

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

● toka-farid@localhost:~/Ansible-labs/day1$ ansible-playbook playbook_task4.yml
BECOME password:

PLAY [playbook_vars] *****

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

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

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

○ toka-farid@localhost:~/Ansible-labs/day1$
```

inventory level:

```
day1 > inventory.ini
1 [web_servers]
2 172.17.0.2
3
4 [web_servers:vars]
5 package_name=nginx
6 package_version=latest
7
```

```
! playbook.yml index.html inventory.ini ! playbook_task3.yml ! playbook_task4.yml x inventory ansible.cfg
day1 > ! playbook_task4.yml > [ ] tasks > { } apt > state
1 - name: My Playbook
2   hosts: web_servers
3
4   tasks:
5     - name: Install Nginx
6       apt:
7         name: "{{ hostvars[inventory_hostname]['package_name'] }}"
8         state: "{{ hostvars[inventory_hostname]['package_version'] }}"
9
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
● toka-farid@localhost:~/Ansible-labs/day1$ ansible-playbook playbook_task4.yml -i inventory.ini
BECOME password:

PLAY [My Playbook] *****

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

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

PLAY RECAP *****
172.17.0.2 : ok=2 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
○ toka-farid@localhost:~/Ansible-labs/day1$
```

command line level:

```
! playbook.yml  index.html  inventory.ini  ! playbook_task3.yml  ! playbook_task4.yml x  inventory  ansible.cfg

day1 > ! playbook_task4.yml > [ ] tasks > { } debug > msg
1  - name: playbook_vars
2    hosts: web_servers
3    vars:
4      package_name: nginx
5      package_version: latest
6
7    tasks:
8      - name: Install Nginx
9        debug:
10         msg: "{{ package_name, package_version }}"
11

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

● toka-farid@localhost:~/Ansible-labs/day1$ ansible-playbook playbook_task4.yml -e "package_name=nginx package_version=latest"
BECOME password:

PLAY [playbook_vars] *****

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

TASK [Install Nginx] *****
ok: [172.17.0.2] => {
  "msg": "('nginx', 'latest')"}

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

○ toka-farid@localhost:~/Ansible-labs/day1$
```