

Task 1:

Write your first playbook file

Stop gather_facts and update cache

```
playbook.yml x
day2 > playbook.yml
1 ---
2 - name: playbook-day2
3   hosts: web_servers
4   gather_facts: False
5
6   tasks:
7     - name: apt update
8       apt:
9         update_cache: True
10
11    - name: copy index file
12      copy:
13        src: ./index.html
14        dest: /var/www/html/index.html
15
16    - name: Restarting the web-server
17      sysvinit:
18        name: nginx
19        state: restarted
```

Task2:

Update cache

Install latest nginx

Copy index.html from controller to host 1

Restart nginx service

```
playbook.yml x
day2 > playbook.yml
1 ---
2 - name: playbook-day2
3   hosts: web_servers
4   gather_facts: False
5
6   tasks:
7     - name: apt update
8       apt:
9         update_cache: True
10
11    - name: copy index file
12      copy:
13        src: ./index.html
14        dest: /var/www/html/index.html
15
16    - name: Restarting the web-server
17      sysvinit:
18        name: nginx
19        state: restarted
```

Task3:

Write simple playbook file

Add two tasks (apt update – apt install nginx)

Add tags to first task: update

Add tags to second task: install

Run only the (apt update) task

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

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

```
playbook-tags.yml x
day2 > playbook-tags.yml
1  ---
2  - name: Test Vars
3    hosts: localhost #web_servers
4    gather_facts: False
5    become: True
6    tasks:
7      - name: Update apt cache
8        tags: update
9        apt:
10         update_cache: True
11
12     - name: Install the latest version of Apache
13       tags: install
14       apt:
15         name: httpd
16         state: latest
17
18     - name: copy index file
19       tags: always
20       copy:
21         src: ./index.html
22         dest: /var/www/html/index.html
23
24
```

Task 4:

Define these variables (package_name, package_version)

on playbook level

on inventory level

on command line level

Use apt module with the package name and version from your variables

— Vars on Playbook level:

```

playbook-vars.yml
---
- name: Test Vars
  hosts: localhost #web_servers
  gather_facts: false
  become: true
  vars:
    pkg_name: httpd
    pkg_version: latest
    cache: true

  tasks:
    - name: Update apt cache
      apt:
        update_cache: "{{ cache }}"

    - name: Install the latest version of Apache
      apt:
        name: "{{ pkg_name }}"
        state: "{{ pkg_version }}"

```

— Vars on inventory level

```

inventory
[web_servers]
172.17.0.2

[vars]
pkg_name = httpd
pkg_version = latest
cache = true

```

```

playbook-vars.yml x  playbook-tags.yml m  pb_when.yml u  test_
day2 > playbook-vars.yml
1  ---
2  - name: Test Vars
3    hosts: localhost #web_servers
4    gather_facts: False
5    become: True
6    vars:
7      p_name: pkg_name
8      p_version: pkg_version
9      u_cache: cache
10
11   tasks:
12   - name: Update apt cache
13     apt:
14       update_cache: "{{ u_cache }}"
15
16   - name: Install the latest version of Apache
17     apt:
18       name: "{{ p_name }}"
19       state: "{{ p_version }}"
20

```

— Vars on command line level

```
day2 > playbooks-vars.yml
1 ---
2 - name: Test Vars
3   hosts: localhost #web_servers
4   gather_facts: False
5   become: True
6   vars:
7     p_name: pkg_name
8     p_version: pkg_version
9     u_cache: cache
10
11   tasks:
12 - name: Update apt cache
13   apt:
14     update_cache: "{{ u_cache }}"
15
16 - name: Install the latest version of Apache
17   apt:
18     name: "{{ p_name }}"
19     state: "{{ p_version }}"
20
21
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
● aalaaseif@aalaaseif:~/ansible-labs$ cd day2
○ aalaaseif@aalaaseif:~/ansible-labs/day2$ ansible-playbook playbooks-vars.yml --e "pkg_name=httpd
pkg_version=latest
cache=True"
```