

Part 1: Tags

- 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`

The screenshot shows a VS Code editor with three files open: `playbook.yml`, `inventory`, and `ansible.cfg`. The `playbook.yml` file contains two tasks: `task1` (tagged `update`) and `task2` (tagged `install`). The `inventory` file lists the host `172.17.0.2`. The `ansible.cfg` file shows the default configuration. The terminal window shows the command `ansible-playbook playbook.yml --tags update` being executed. The output indicates that the `task1` was successful, with the `apt` package being updated on `172.17.0.2`.

```
playbook.yml
1 - name: play1
2   gather_facts: false
3   hosts: 172.17.0.2
4
5   tasks:
6     - name: task1
7       tags: update
8       apt:
9         update_cache: true
10
11    - name: task2
12      tags: install
13      apt:
14        name: nginx
15        state: latest

inventory
1 172.17.0.2

ansible.cfg
1 [defaults]
2 inventory = ./inventory
3 private_key_file = ~/.ssh/key
4 remote_user = ansible
5
6 [privilege_escalation]
7 become = true
8 become_ask_pass = true
9
10 # become --> to run as sudo

terminal
spot@spot-pc:~/Downloads/day 2/Lab 2/part1$ ansible-playbook playbook.yml --tags update
BECOME password:

PLAY [play1] *****

TASK [task1] *****
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

spot@spot-pc:~/Downloads/day 2/Lab 2/part1$
```

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

The screenshot shows the same VS Code editor with the `playbook2.yml` file added. This file contains a new task `task2` with the tag `always`. The terminal window shows the command `ansible-playbook playbook2.yml --tags update` being executed. The output indicates that both `task1` and `task2` were successful, with the `apt` package being updated on `172.17.0.2`.

```
playbook2.yml
1 - name: play1
2   gather_facts: false
3   hosts: 172.17.0.2
4
5   tasks:
6     - name: task1
7       tags: update
8       apt:
9         update_cache: true
10
11    - name: task2
12      tags: [always]
13      apt:
14        name: nginx
15        state: latest

inventory
1 172.17.0.2

ansible.cfg
1 [defaults]
2 inventory = ./inventory
3 private_key_file = ~/.ssh/key
4 remote_user = ansible
5
6 [privilege_escalation]
7 become = true
8 become_ask_pass = true
9
10 # become --> to run as sudo

terminal
spot@spot-pc:~/Downloads/day 2/Lab 2/part1$ ansible-playbook playbook2.yml --tags update
BECOME password:

PLAY [play1] *****

TASK [task1] *****
ok: [172.17.0.2]

TASK [task2] *****
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

spot@spot-pc:~/Downloads/day 2/Lab 2/part1$
```

Part 2: Variables

- 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

The image consists of three vertically stacked screenshots of the Visual Studio Code editor, demonstrating Ansible workflow and variable usage.

Top Screenshot: Shows the Explorer sidebar with a workspace containing 'Lab 1' and 'Lab 2'. 'Lab 2' contains 'part1 (tags)' and 'part2 (variables)'. The main editor shows 'playbook.yml' with the following content:

```
1 - name: play1
2   gather_facts: false
3   hosts: 172.17.0.2
4   vars:
5     package_name: nginx
6     package_state: latest
7
8   tasks:
9     - name: task1
10      tags: update
11      apt:
12        name: "{{ package_name }}"
13        state: "{{ package_state }}"
```

The TERMINAL panel shows the execution of `ansible-playbook playbook.yml`. The output indicates success for host 172.17.0.2.

Middle Screenshot: Shows the same workspace, but the main editor now displays 'inventory' with the following content:

```
1 172.17.0.2 package_name=nginx package_state=latest
```

The TERMINAL panel shows the execution of `ansible-playbook playbook2.yml`, which also succeeds.

Bottom Screenshot: Shows the workspace with 'playbook2.yml' in the main editor. The content is identical to 'playbook.yml' but uses variables for the package name and state:

```
1 - name: play1
2   gather_facts: false
3   hosts: 172.17.0.2
4
5   tasks:
6     - name: task1
7       tags: update
8       apt:
9         name: "{{ package_name }}"
10        state: "{{ package_state }}"
```

The TERMINAL panel shows the execution of `ansible-playbook playbook2.yml -e "package_name=nginx package_state=latest"`, which succeeds.

Part 3: Loops

- Loop over a list of packages and install latest versions.
- Loop over a list of packages and perform different actions as per input.

The screenshot shows the VS Code interface with the Explorer panel on the left displaying the file structure of a workspace. The main editor shows the `playbook.yml` file with the following content:

```
1 - name: play1
2   gather_facts: false
3   hosts: 172.17.0.2
4
5   tasks:
6     - name: task1
7       apt:
8         name: "{{ item }}"
9         state: latest
10      loop:
11        - nginx
12        - curl
13        - ssh
14        - sudo
```

The terminal panel at the bottom shows the output of the command `ansible-playbook playbook.yml`. It displays the standard Ansible output, including the play name, task name, and the results of the `apt` module for each package in the loop.

```
spot@spot-pc:~/Downloads/day 2/Lab 2/part3 (loops)$ ansible-playbook playbook.yml
BECOME password:

PLAY [play1] *****

TASK [task1] *****
ok: [172.17.0.2] => (item=nginx)
changed: [172.17.0.2] => (item=curl)
ok: [172.17.0.2] => (item=ssh)
ok: [172.17.0.2] => (item=sudo)

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

The screenshot shows the VS Code interface with the Explorer panel on the left displaying the file structure of a workspace. The main editor shows the `playbook2.yml` file with the following content:

```
1 - name: play1
2   gather_facts: false
3   hosts: 172.17.0.2
4   vars:
5     my_list:
6       - nginx
7       - curl
8       - ssh
9       - sudo
10
11   tasks:
12     - name: task1
13       apt:
14         name: "{{ item }}"
15         state: latest
16       loop: "{{ my_list }}"
```

The terminal panel at the bottom shows the output of the command `ansible-playbook playbook2.yml`. It displays the standard Ansible output, including the play name, task name, and the results of the `apt` module for each package in the loop.

```
spot@spot-pc:~/Downloads/day 2/Lab 2/part3 (loops)$ ansible-playbook playbook2.yml
BECOME password:

PLAY [play1] *****

TASK [task1] *****
ok: [172.17.0.2] => (item=nginx)
ok: [172.17.0.2] => (item=curl)
ok: [172.17.0.2] => (item=ssh)
ok: [172.17.0.2] => (item=sudo)

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

Part 4: When

- Install nginx or apache2 depending on distribution
- Restart nginx service if distribution is ubuntu and variable value is true

```
EXPLORER  ...  ≡ playbook.yml ●
Lab 2 > part4 (when) > ≡ playbook.yml
> Lab 1
> Lab 2
  > part1 (tags)
  > part2 (variables)
  > part3 (loops)
  > part4 (when)
    ansible.cfg
    inventory
    playbook.yml
    ~/.lock.Untitled 1.o...
    Untitled 1.odt

1  - name: my play with conditions
2  hosts: 172.17.0.2
3  vars:
4    var1: true
5  tasks:
6  - name: task1 with conditions
7    package:
8      name: nginx
9    when: ansible_facts['distribution'] == "Ubuntu"
10 - name: task 2 with conditions
11   package:
12     name: httpd
13   when: ansible_facts['distribution'] == "CentOS"
14 - name: task 3 Restart nginx if os is Ubuntu
15   sysvinit:
16     name: nginx
17     state: restarted
18     enabled: yes
19   when: ansible_facts['distribution'] == "Ubuntu"
20 - name: task 4 Restart nginx on Ubuntu if var1 = true
21   sysvinit:
22     name: nginx
23     state: restarted
24     enabled: yes
25   when: var1 == true
```

```
spot@spot-pc:~/Downloads/day 2/Lab 2/part4 (when)$ ansible-playbook playbook.yml
BECOME password:

PLAY [my play with conditions] *****

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

TASK [task1 with conditions] *****
ok: [172.17.0.2]

TASK [task 2 with conditions] *****
skipping: [172.17.0.2]

TASK [task 3 Restart nginx if os is Ubuntu] *****
changed: [172.17.0.2]

TASK [task 4 Restart nginx on Ubuntu if var1 = true] *****
changed: [172.17.0.2]

PLAY RECAP *****
172.17.0.2 : ok=4  changed=2  unreachable=0  failed=0  skipped=1  rescued=0  ignored=0

spot@spot-pc:~/Downloads/day 2/Lab 2/part4 (when)$
```

Part 5: Register & When

- View the value of your register variable using debug module

```
Lab 2 > part5 (register) > playbook.yml
1 - name: my play with conditions
2   gather_facts: false
3   hosts: 172.17.0.2
4
5   tasks:
6     - name: task1
7       command: cat /var/www/html/index.html
8       register: my_result
9     - name: task 2
10      debug:
11        var: my_result
```

```
spot@spot-pc:~/Downloads/day 2/Lab 2/part5 (register)$ ansible-playbook playbook.yml
BECOME password:

PLAY [my play with conditions] *****

TASK [task1] *****
changed: [172.17.0.2]

TASK [task 2] *****
ok: [172.17.0.2] => {
  "my_result": {
    "ansible_facts": {
      "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": true,
    "cmd": [
      "cat",
      "/var/www/html/index.html"
    ],
    "delta": "0:00:00.007580",
    "end": "2023-05-04 16:16:24.628119",
    "failed": false,
    "msg": "",
    "rc": 0,
    "start": "2023-05-04 16:16:24.620539",
    "stderr": "",
    "stderr_lines": [],
    "stdout": "Hello world",
    "stdout_lines": [
      "Hello world"
    ]
  }
}

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

spot@spot-pc:~/Downloads/day 2/Lab 2/part5 (register)$
```

- Restart service if the installation task was changed or was not failed

```
Lab 2 > part5 (register) > playbook2.yml
1 - name: RegisterAndWhen
2   gather_facts: false
3   hosts: 172.17.0.2
4   tasks:
5     - name: task1
6       package:
7         name: nginx
8         state: present
9         register: nginx_install
10    - name: task2
11      sysvinit:
12        name: nginx
13        state: restarted
14        enabled: yes
15      when: nginx_install.changed or not nginx_install.failed
```

```
spot@spot-pc:~/Downloads/day 2/Lab 2/part5 (register)$ ansible-playbook playbook2.yml
BECOME password:

PLAY [RegisterAndWhen] *****

TASK [task1] *****
ok: [172.17.0.2]

TASK [task2] *****
changed: [172.17.0.2]

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

spot@spot-pc:~/Downloads/day 2/Lab 2/part5 (register)$
```

Part 6: Templates

The screenshot shows a VS Code workspace with two files: `playbook.yml` and `index.html.j2`. The `playbook.yml` file is open in the main editor, showing a playbook named `templates` with a task `task1` that uses the `template` module to copy `index.html.j2` to `/var/www/html/index.html` on host `172.17.0.2`. The `index.html.j2` file is also open, showing a single line of Jinja2 template code: `{{ welcome_msg }}`.

The terminal at the bottom shows the execution of the playbook. It starts with a `curl` command to check the host, followed by `ansible-playbook playbook.yml`. The output shows the playbook running successfully on `172.17.0.2`, with the `template` task changing the file.

```
1 - name: templates
2   gather_facts: false
3   hosts: 172.17.0.2
4   vars:
5     welcome_msg: Hello template from playbook
6
7   tasks:
8     - name: task1
9       template:
10         src: ./index.html.j2
11         dest: /var/www/html/index.html
```

```
1 {{ welcome_msg }}
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** POLYGLOT NOTEBOOK GITLENS

```
● Hello worldspot@spot-pc:~/Downloads/day 2/Lab 2/part6 (templates)$ curl 172.17.0.2
● Hello worldspot@spot-pc:~/Downloads/day 2/Lab 2/part6 (templates)$ ansible-playbook playbook.yml
BECOME password:

PLAY [templates] *****

TASK [task1] *****
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
```

Part 7: Handlers

EXPLORER

...
UNTITLED (WORKSPACE)
Lab 1
Lab 2
part1 (tags)
part2 (variables)
part3 (loops)
part4 (when)
part5 (register)
part6 (templates)
part7 (handlers)
ansible.cfg
inventory
playbook.yml
lock.Untitled 1.o...
Untitled 1.odt

playbook.yml
Lab 2 > part7 (handlers) > playbook.yml
1 - name: my play with handlers
2 gather_facts: false
3 hosts: 172.17.0.2
4
5 tasks:
6 - name: my task1 with handlers
7 apt:
8 name: nginx
9 notify: my_handler
10
11 handlers:
12 - name: my_handler
13 sysvinit:
14 name: nginx
15 state: restarted
16 enabled: yes

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL POLYGLOT NOTEBOOK GITLENS
spot@spot-pc:~/Downloads/day 2/Lab 2/part7 (handlers)\$ ansible-playbook playbook.yml
BECOME password:

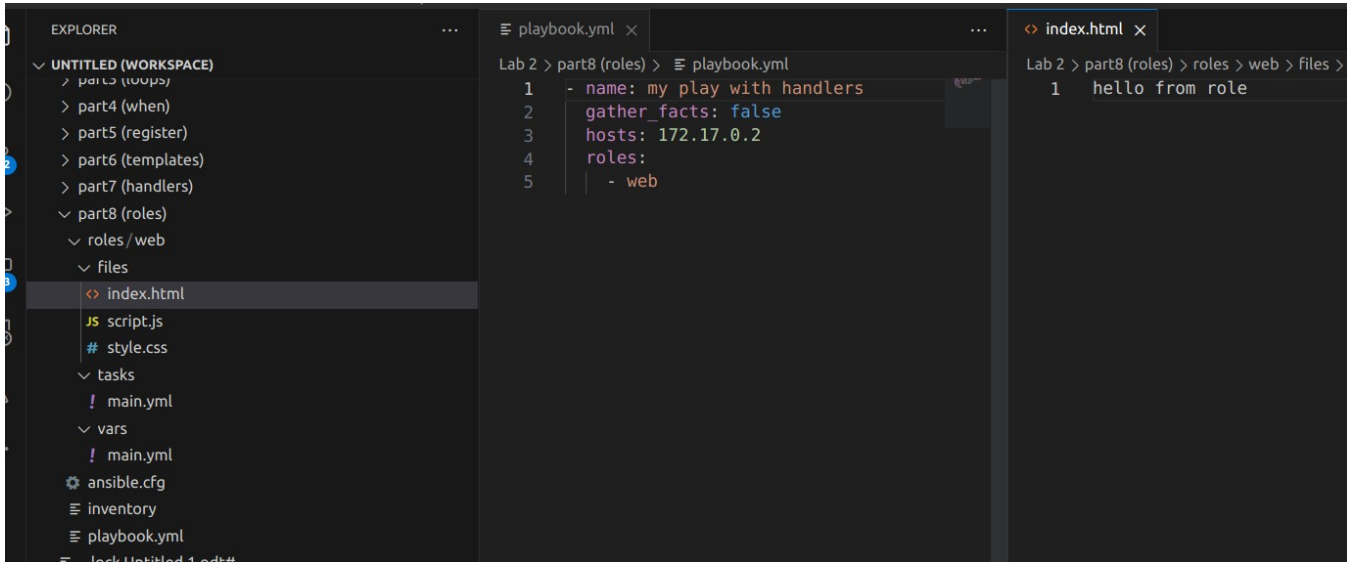
PLAY [my play with handlers]

TASK [my task1 with handlers]
ok: [172.17.0.2]

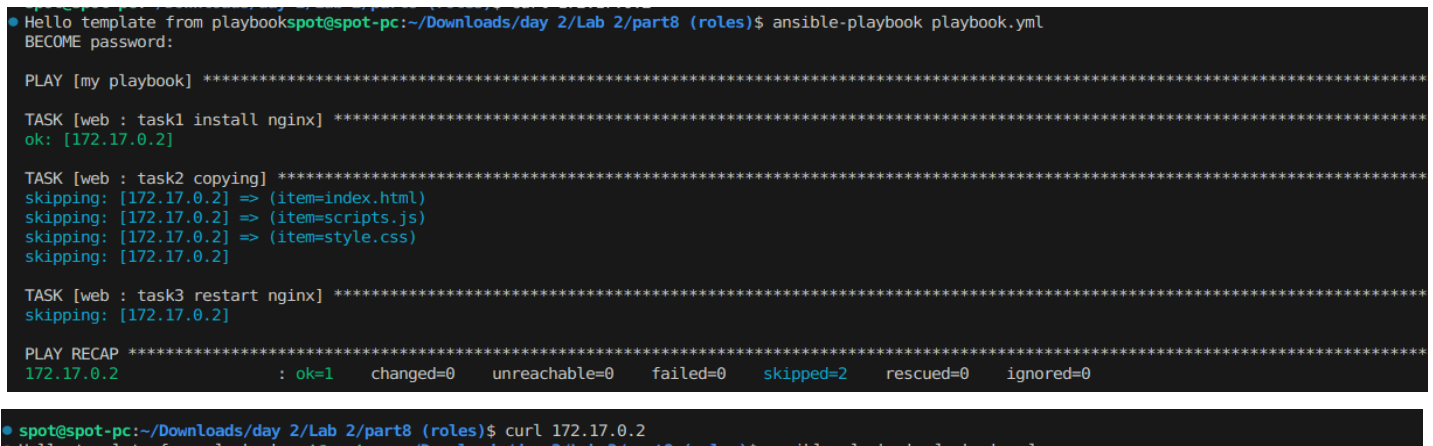
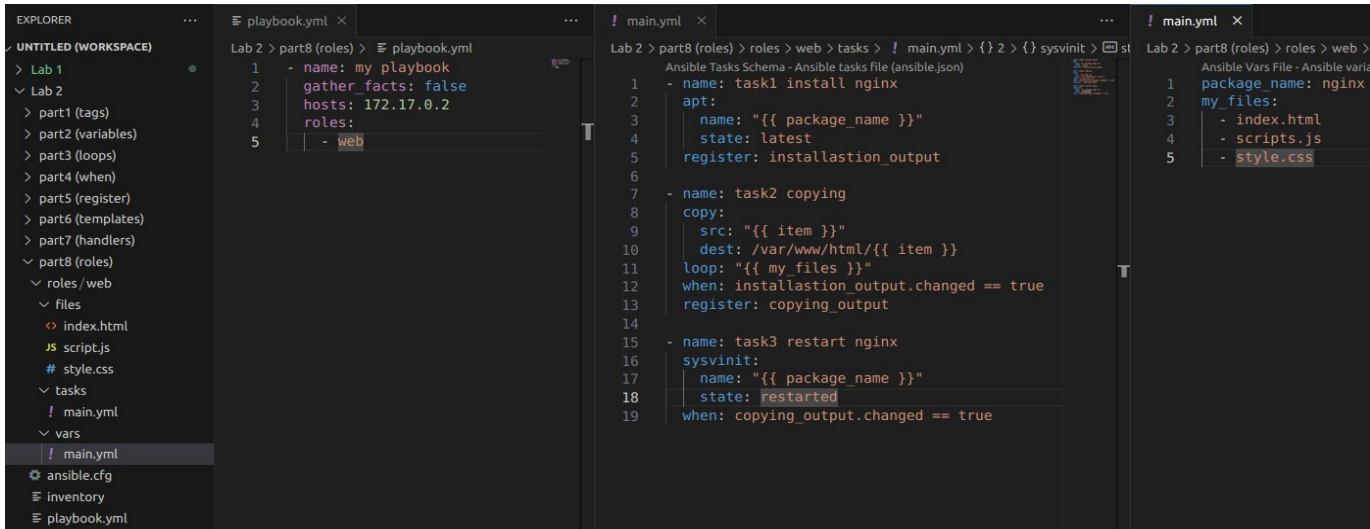
PLAY RECAP
172.17.0.2 : ok=1 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

Part 8: Roles

- Create your first role with name (web)



- The task book will include:
 - installing a package
(get the package name from vars)
 - copying a list of files from controller to host using loop
(get the list of file names from vars)
(the actual files will be stored in `./roles/web/files`)
(will be executed only when the install task is in state: changed)
 - Restart the service of the installed package
(will be executed only when the copy task is in state: changed)



- using Handlers

The screenshot shows a VS Code editor with three tabs: `main.yml`, `main.yml`, and `index.html`. The first `main.yml` tab shows a task `task1 install nginx` with a `notify: handler copying` directive. The second `main.yml` tab shows a handler `handler copying` that copies files to `/var/www/html/{{ item }}` and a `handler restart` that restarts the `nginx` service. The `index.html` tab shows a simple `hello from role (using handlers)` message. The terminal output shows the execution of the playbook, including the `PLAY RECAP` and the `PLAY [my playbook]` section, which shows the `handler copying` and `handler restart` handlers being executed.

```
1 - name: task1 install nginx
2   sysvinit:
3     name: nginx
4     state: restarted
5     enabled: yes
6     notify: handler copying
7
8   # handler runs only if state of task is changed
9
10
11
12
```

```
1 - name: handler copying
2   copy:
3     src: "{{ item }}"
4     dest: /var/www/html/{{ item }}
5     loop: "{{ my_files }}"
6     notify: handler restart
7
8 - name: handler restart
9   sysvinit:
10    name: "{{ package_name }}"
11    state: restarted
12
```

```
1 hello from role (using handlers)
```

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

spot@spot-pc:~/Downloads/day 2/Lab 2/part8 (roles)\$ curl 172.17.0.2
spot@spot-pc:~/Downloads/day 2/Lab 2/part8 (roles)\$ curl 172.17.0.2
hello from role (using handlers)
spot@spot-pc:~/Downloads/day 2/Lab 2/part8 (roles)\$ ansible-playbook playbook.yml
BECOME password:

PLAY [my playbook] *****

TASK [web : task1 install nginx] *****
changed: [172.17.0.2]

RUNNING HANDLER [web : handler copying] *****
changed: [172.17.0.2] => (item=index.html)
changed: [172.17.0.2] => (item=script.js)
ok: [172.17.0.2] => (item=style.css)

RUNNING HANDLER [web : handler restart] *****
^[[A^[[A^[[B^[[Bchanged: [172.17.0.2]

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

spot@spot-pc:~/Downloads/day 2/Lab 2/part8 (roles)\$ curl 172.17.0.2
hello from role (using handlers)spot@spot-pc:~/Downloads/day 2/Lab 2/part8 (roles)\$

- using templates also

The screenshot shows a VS Code editor with four tabs: `main.yml`, `main.yml`, `index.html.j2`, and `main.yml`. The first `main.yml` tab shows a task `task1 install nginx` with a `notify: handler copying` directive. The second `main.yml` tab shows a handler `handler copying` that copies files to `/var/www/html/{{ item }}` and a `handler restart` that restarts the `nginx` service. The `index.html.j2` tab shows a Jinja2 template `{{ welcome_msg }}`. The `main.yml` tab shows a handler `template copying` that copies the `index.html.j2` template to `/var/www/html/index.html` and a `handler restart` that restarts the `nginx` service. The terminal output shows the execution of the playbook, including the `PLAY RECAP` and the `PLAY [my playbook]` section, which shows the `handler copying`, `template copying`, and `handler restart` handlers being executed.

```
1 - name: task1 install nginx
2   sysvinit:
3     name: nginx
4     state: restarted
5     enabled: yes
6     notify: handler copying
7
8   # handler runs only if state of task
9
10
11
12
```

```
1 package name: nginx
2 welcome_msg: Hello from the template
3 my_files:
4   - index.html
5   - script.js
6   - style.css
7
```

```
1 {{ welcome_msg }}
```

```
1 - name: handler copying
2   copy:
3     src: "{{ item }}"
4     dest: /var/www/html/{{ item }}
5     loop: "{{ my_files }}"
6     notify: template copying
7
8 - name: template copying
9   template:
10    src: index.html.j2
11    dest: /var/www/html/index.html
12    notify: handler restart
13
14 - name: handler restart
15   sysvinit:
16    name: "{{ package_name }}"
17    state: restarted
18    enabled: yes
19
```

172.17.0.2 : ok=3 changed=3 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

spot@spot-pc:~/Downloads/day 2/Lab 2/part8 (roles)\$ curl 172.17.0.2
Hello from the templatespot@spot-pc:~/Downloads/day 2/Lab 2/part8 (roles)\$ ansible-playbook playbook.yml
BECOME password:

PLAY [my playbook] *****

TASK [web : task1 install nginx] *****
changed: [172.17.0.2]

RUNNING HANDLER [web : handler copying] *****
changed: [172.17.0.2] => (item=index.html)
ok: [172.17.0.2] => (item=script.js)
ok: [172.17.0.2] => (item=style.css)

RUNNING HANDLER [web : template copying] *****
changed: [172.17.0.2]

RUNNING HANDLER [web : handler restart] *****
changed: [172.17.0.2]

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

spot@spot-pc:~/Downloads/day 2/Lab 2/part8 (roles)\$ curl 172.17.0.2
spot@spot-pc:~/Downloads/day 2/Lab 2/part8 (roles)\$ curl 172.17.0.2
Hello from the templatespot@spot-pc:~/Downloads/day 2/Lab 2/part8 (roles)\$