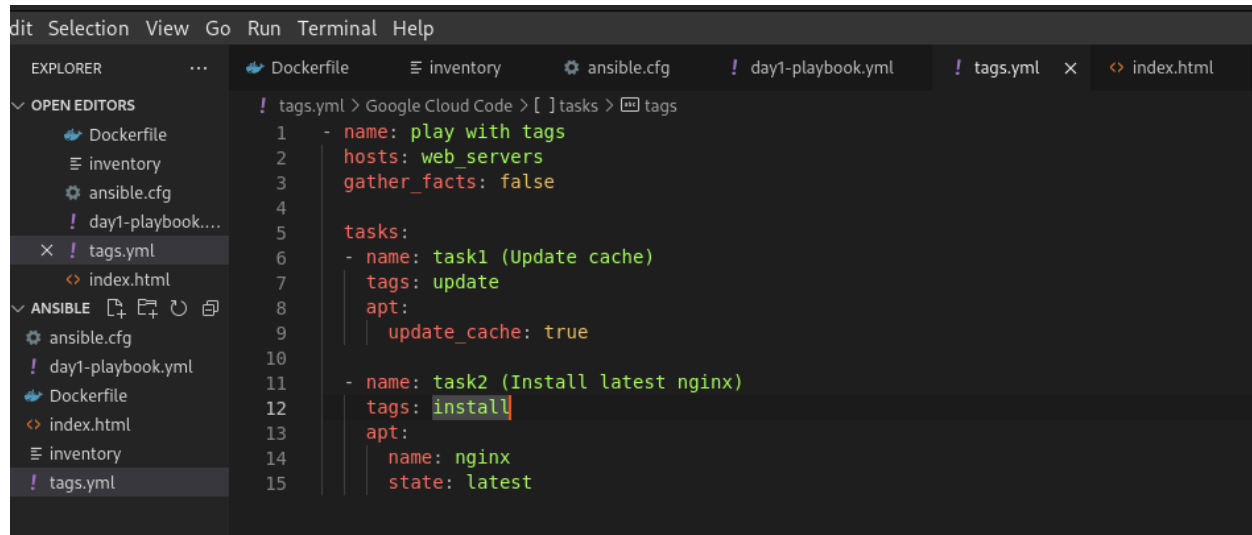
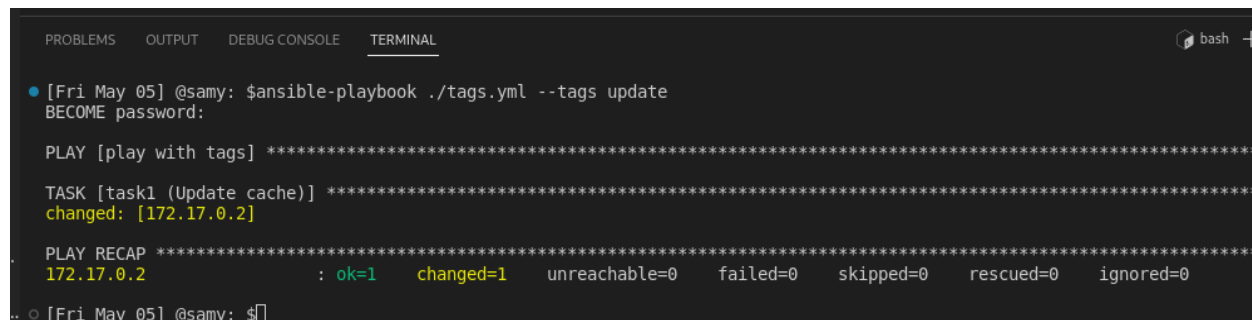


- ▶ Write simple playbook file
- ▶ Add two tasks (apt update – apt install nginx)
- ▶ Add tags to first task: update
- ▶ Add tags to second task: install
- ▶ Run only the (apt update) task
- ▶ Example: `ansible-playbook my-playbook.yml --tags update`
- ▶ Add one task with "tags: always" and run the previous command again

## TAGS



```
1 - name: play with tags
2   hosts: web_servers
3   gather_facts: false
4
5   tasks:
6     - name: task1 (Update cache)
7       tags: update
8       apt:
9         update_cache: true
10
11     - name: task2 (Install latest nginx)
12       tags: install
13       apt:
14         name: nginx
15         state: latest
```



```
[Fri May 05] @samy: $ansible-playbook ./tags.yml --tags update
BECOME password:

PLAY [play with tags] *****

TASK [task1 (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

.. [Fri May 05] @samy: $
```

```
Dockerfile  inventory  ansible.cfg  ! day1-playbook.yml  ! tags.yml  x  < index.html

! tags.yml > Google Cloud Code > [ ] tasks > tags
1  - name: play with tags
2    hosts: web_servers
3    gather_facts: false
4
5    tasks:
6      - name: task1 (Update cache)
7        tags: update
8        apt:
9          update_cache: true
10
11      - name: task2 (Install latest nginx)
12        tags: install, always
13        apt:
14          name: nginx
15          state: latest

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

• [Fri May 05] @samy: $ansible-playbook ./tags.yml --tags update
BECOME password:

PLAY [play with tags] *****

TASK [task1 (Update cache)] *****
ok: [172.17.0.2]

TASK [task2 (Install latest 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
```

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

## VARIABLES

==> on playbook level

The screenshot shows the VS Code editor with the file explorer on the left. The 'variables.yml' file is open, displaying an Ansible playbook. The breadcrumb at the top indicates the path: `variables.yml > Google Cloud Code > [ ] tasks > { } apt > state`. The playbook content is as follows:

```
1 - name: play with variables
2   hosts: web_servers
3   gather_facts: false
4   vars:
5     package_name: nginx
6     package_version: latest
7   tasks:
8     - name: task1 (Install latest nginx)
9       apt:
10         name: "{{ package_name }}"
11         state: "{{ package_version }}"
```

The screenshot shows the terminal window with the following output:

```
[Fri May 05] @samy: $ansible-playbook ./variables.yml
BECOME password:

PLAY [play with variables] *****

TASK [task1 (Install latest nginx)] *****
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
```

==> on inventory level

The screenshot shows the VS Code editor with the file explorer on the left. The 'inventory' file is open, displaying the inventory content. The breadcrumb at the top indicates the path: `inventory > ansible.cfg > day1-playbook.yml > tags.yml > variables.yml`. The inventory content is as follows:

```
1 [web_servers]
2 172.17.0.2 package_name=nginx package_version=latest
3
4 [database_servers]
5 3.87.24.252
6 3.87.24.253
7 3.87.24.252
8
9 [database_servers:vars]
10 package_name=nginx
11 package_version=latest
12
13 [other_groub]
14 172.17.0.5
```

```
EXPLORER  ...  inventory  ansible.cfg  ! day1-playbook.yml  ! tags.yml  ! variables.yml x
v OPEN EDITORS
  inventory
  ansible.cfg
  ! day1-playbook....
  ! tags.yml
  x ! variables.yml
v ANSIBLE
  ansible.cfg
  ! day1-playbook.yml
  Dockerfile
  index.html
  inventory
  ! tags.yml
  ! variables.yml

! variables.yml > YAML > {} 0
1  - name: play with variables
2    hosts: web_servers
3    gather_facts: false
4    # vars:
5    #   package_name: nginx
6    #   package_version: latest
7    tasks:
8      - name: task1 (Install latest nginx)
9        apt:
10          name: "{{ package_name }}"
11          state: "{{ package_version }}"

[Fri May 05] @samy: $ansible-playbook ./variables.yml
BECOME password:

PLAY [play with variables] *****

TASK [task1 (Install latest nginx)] *****
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
```

==> on command line level

```
[Fri May 05] @samy: $ansible-playbook ./variables.yml -e "package_name=nginx package_version=latest"
BECOME password:

PLAY [play with variables] *****

TASK [task1 (Install latest nginx)] *****
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
```

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

## LOOPS

==> Loop over a list of packages and install latest versions.

```
! loops.yml > Google Cloud Code > [ ] loop
1 - name: play with loops
2   hosts: web_servers
3   gather_facts: false
4
5   tasks:
6   - name: my task with loops
7     apt:
8       name: "{{ item }}"
9       state: latest
10    loop:
11      - nginx
12      - curl
13      - ssh
14      - sudo
```

```
• [Fri May 05] @samy: $ansible-playbook ./loops.yml
BECOME password:

PLAY [play with loops] *****

TASK [my task with loops] *****
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
```

==> Loop over a list of packages and perform different actions as per input.

```
! loops.yml > Google Cloud Code > [ ] tasks > [ ] loop > package_name
1 - name: play with loops
2   hosts: web_servers
3   gather_facts: false
4
5   tasks:
6   - name: task1
7     apt:
8       name: "{{ item.package_name }}"
9       state: "{{ item.package_state }}"
10    loop:
11      - { package_name: "nginx", package_state: "present" }
12      - { package_name: "curl", package_state: "absent" }
13      - { package_name: "ssh", package_state: "present" }
14
15 # tasks:
```

```
• [Sat May 06] @samy: $ansible-playbook ./loops.yml
BECOME password:

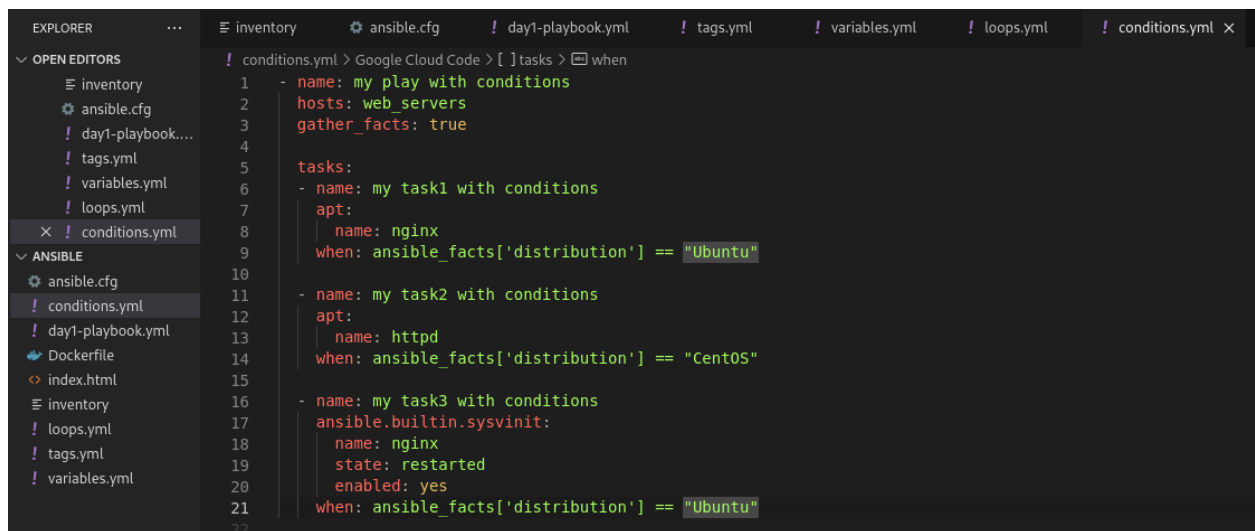
PLAY [play with loops] *****

TASK [task1] *****
ok: [172.17.0.2] => (item={'package_name': 'nginx', 'package_state': 'present'})
ok: [172.17.0.2] => (item={'package_name': 'curl', 'package_state': 'absent'})
ok: [172.17.0.2] => (item={'package_name': 'ssh', 'package_state': 'present'})

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

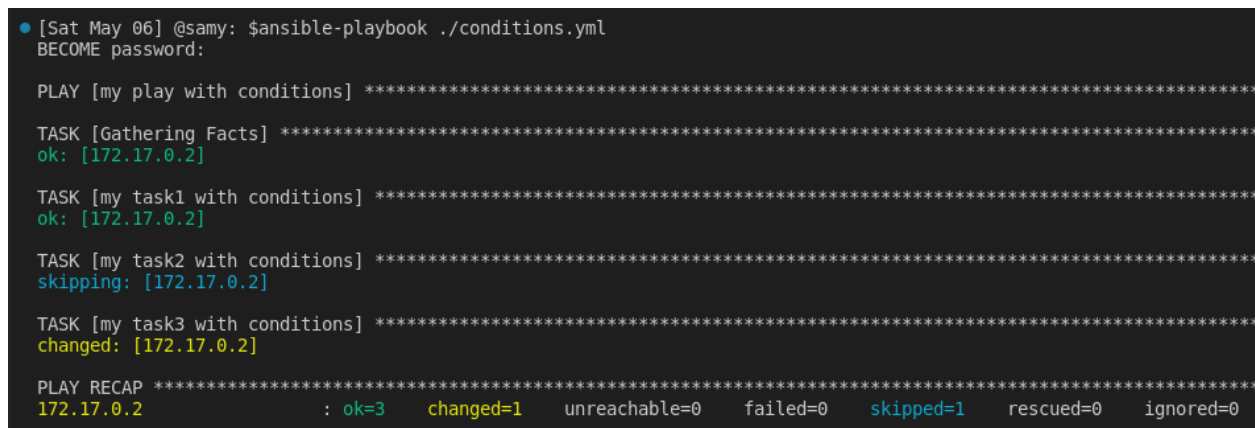
- ▶ Install nginx or apache2 depending on distribution
- ▶ Restart nginx service if distribution is ubuntu and variable value is true

## WHEN



The screenshot shows the VS Code interface with an Ansible playbook named `conditions.yml` open. The left sidebar shows the Explorer view with the file tree. The main editor shows the following YAML code:

```
1 - name: my play with conditions
2   hosts: web_servers
3   gather_facts: true
4
5   tasks:
6     - name: my task1 with conditions
7       apt:
8         name: nginx
9       when: ansible_facts['distribution'] == "Ubuntu"
10
11     - name: my task2 with conditions
12       apt:
13         name: httpd
14       when: ansible_facts['distribution'] == "CentOS"
15
16     - name: my task3 with conditions
17       ansible.builtin.sysvinit:
18         name: nginx
19         state: restarted
20         enabled: yes
21       when: ansible_facts['distribution'] == "Ubuntu"
22
```



The screenshot shows the terminal output of the Ansible playbook execution. The output is as follows:

```
● [Sat May 06] @samy: $ansible-playbook ./conditions.yml
BECOME password:

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

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

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

TASK [my task2 with conditions] *****
skipping: [172.17.0.2]

TASK [my task3 with conditions] *****
changed: [172.17.0.2]

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

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

# REGISTER & WHEN

==> View the value of your register variable using debug module.

The screenshot shows a VS Code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project named 'register.yml' with files: inventory, ansible.cfg, day1-playbook.yml, tags.yml, variables.yml, loops.yml, conditions.yml, and register.yml. The code editor shows the content of 'register.yml' with line numbers 1 through 13. The playbook has two tasks: 'my task1 with register' and 'my task2 with register'. Task 1 uses the 'register' module to store the result of a 'cat' command in a variable named 'my\_result'. Task 2 uses the 'debug' module to display the value of 'my\_result'.

```

1  - name: my play with register
2    hosts: web_servers
3    gather_facts: false
4
5    tasks:
6      - name: my task1 with register
7        command: cat /var/www/html/index.html
8        register: my_result
9
10     - name: my task2 with register
11       debug:
12         var: my_result
13

```

The screenshot shows a terminal window with the output of an Ansible playbook execution. The command executed is '\$ansible-playbook ./register.yml'. The output shows the playbook running on host '172.17.0.2'. Task 1, 'my task1 with register', is successful and changes the state of the host. Task 2, 'my task2 with register', is successful and displays the value of 'my\_result' as a dictionary containing 'ansible\_facts' and 'changed' status. The final output shows the playbook recap with 2 OK, 1 changed, and 0 failed tasks.

```

[Sat May 06] @samy: $ansible-playbook ./register.yml
BECOME password:

PLAY [my play with register] *****

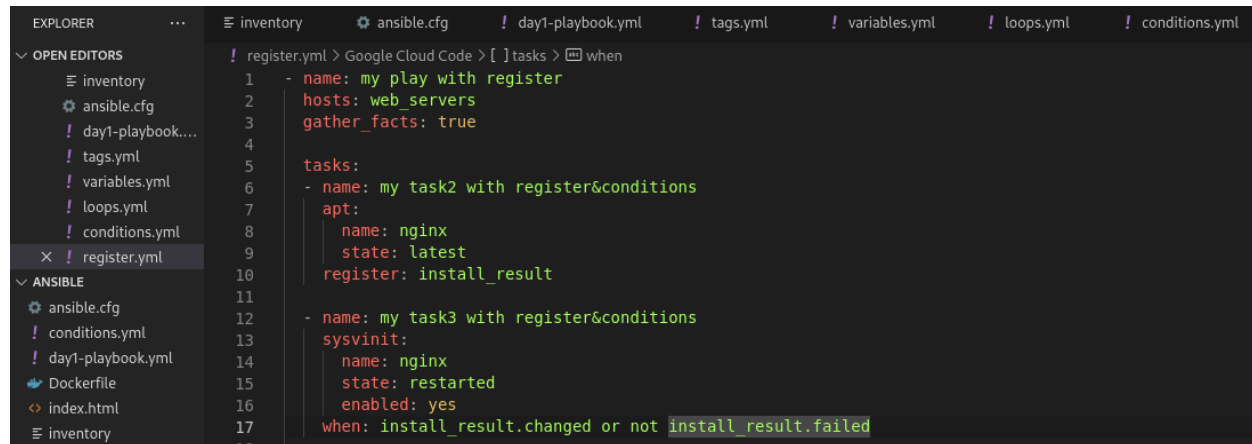
TASK [my task1 with register] *****
changed: [172.17.0.2]

TASK [my task2 with register] *****
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": "0:00:00.002824",
    "end": "2023-05-05 22:33:49.619788",
    "failed": false,
    "msg": "",
    "rc": 0,
    "start": "2023-05-05 22:33:49.616964",
    "stderr": "",
    "stderr_lines": [],
    "stdout": "==== Hello Ansible =====",
    "stdout_lines": [
      "==== Hello Ansible ====="
    ]
  }
}

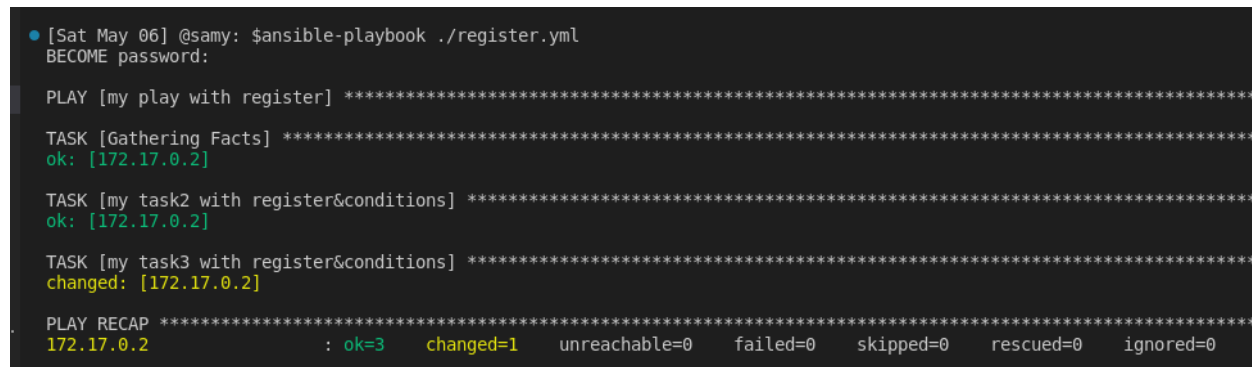
PLAY RECAP *****
172.17.0.2 : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

```

==> Restart service if the installation task was changed or was not failed.



```
1 - name: my play with register
2   hosts: web_servers
3   gather_facts: true
4
5   tasks:
6     - name: my task2 with register&conditions
7       apt:
8         name: nginx
9         state: latest
10        register: install_result
11
12     - name: my task3 with register&conditions
13       sysvinit:
14         name: nginx
15         state: restarted
16         enabled: yes
17       when: install_result.changed or not install_result.failed
```



```
[Sat May 06] @samy: $ansible-playbook ./register.yml
BECOME password:

PLAY [my play with register] *****

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

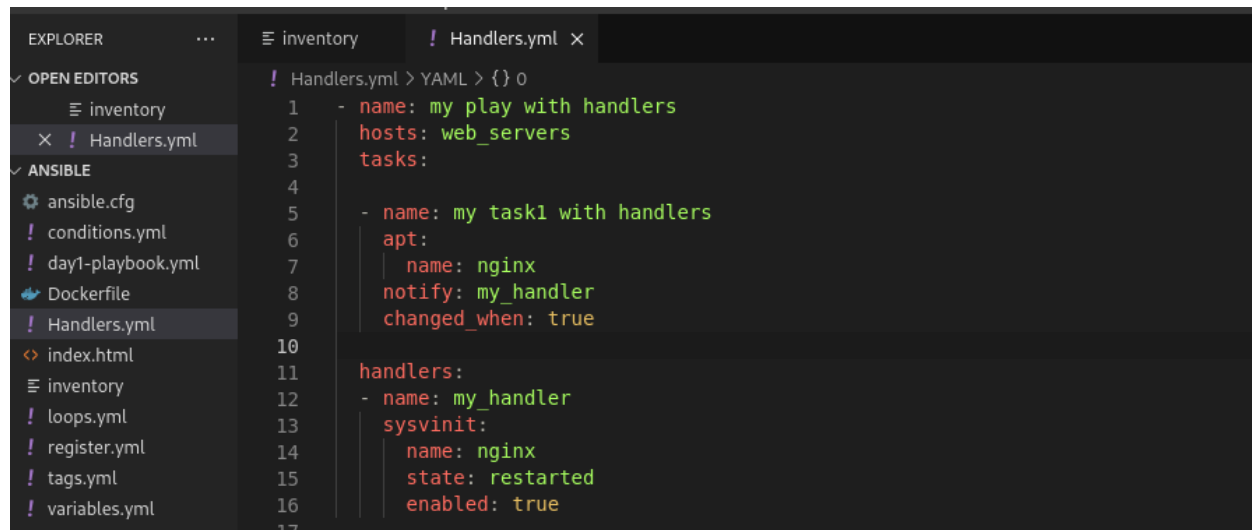
TASK [my task2 with register&conditions] *****
ok: [172.17.0.2]

TASK [my task3 with register&conditions] *****
changed: [172.17.0.2]

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

=====

## =====> HANDLERS <=====



```
1 - name: my play with handlers
2   hosts: web_servers
3   tasks:
4
5     - name: my task1 with handlers
6       apt:
7         name: nginx
8         notify: my_handler
9         changed_when: true
10
11   handlers:
12     - name: my_handler
13       sysvinit:
14         name: nginx
15         state: restarted
16         enabled: true
```



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

• [Sun May 07] @samy: $ansible-playbook ./Handlers.yml
BECOME password:

PLAY [my play with handlers] *****

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

TASK [my task1 with handlers] *****
changed: [172.17.0.2]

RUNNING HANDLER [my_handler] *****
changed: [172.17.0.2]

PLAY RECAP *****
172.17.0.2 : ok=3 changed=2 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
```

## =====>> TEMPLATES <<=====

```
EXPLORER ... inventory ! Handlers.yml ! Templates.yml x index.htmlj2

OPEN EDITORS
  inventory
  ! Handlers.yml
  x ! Templates.yml
  index.htmlj2

ANSIBLE
  ansible.cfg
  ! conditions.yml
  ! day1-playbook.yml
  Dockerfile
  ! Handlers.yml
  < index.html
  index.htmlj2
  inventory

! Templates.yml > Google Cloud Code > {} vars > welcome_msg
1 - name: my play with templates
2   hosts: web_servers
3   gather_facts: false
4   vars:
5     welcome_msg: ===== Hello! this is Ansible course =====
6
7   tasks:
8     - name: task1 (Copy dynamic content from controller to host 1 using templates)
9       template:
10         src: ./index.html.j2
11         dest: /var/www/html/index.html
12
```

```
EXPLORER ... inventory ! Handlers.yml ! Templates.yml index.htmlj2 x

OPEN EDITORS
  inventory
  ! Handlers.yml
  ! Templates.yml
  x index.htmlj2

ANSIBLE
  ansible.cfg
  ! conditions.yml
  ! day1-playbook.yml
  Dockerfile
  ! Handlers.yml
  index.html
  index.htmlj2
  inventory

index.htmlj2
1 {{ welcome_msg }}
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

BECOME password:

PLAY [my play with templates] *****

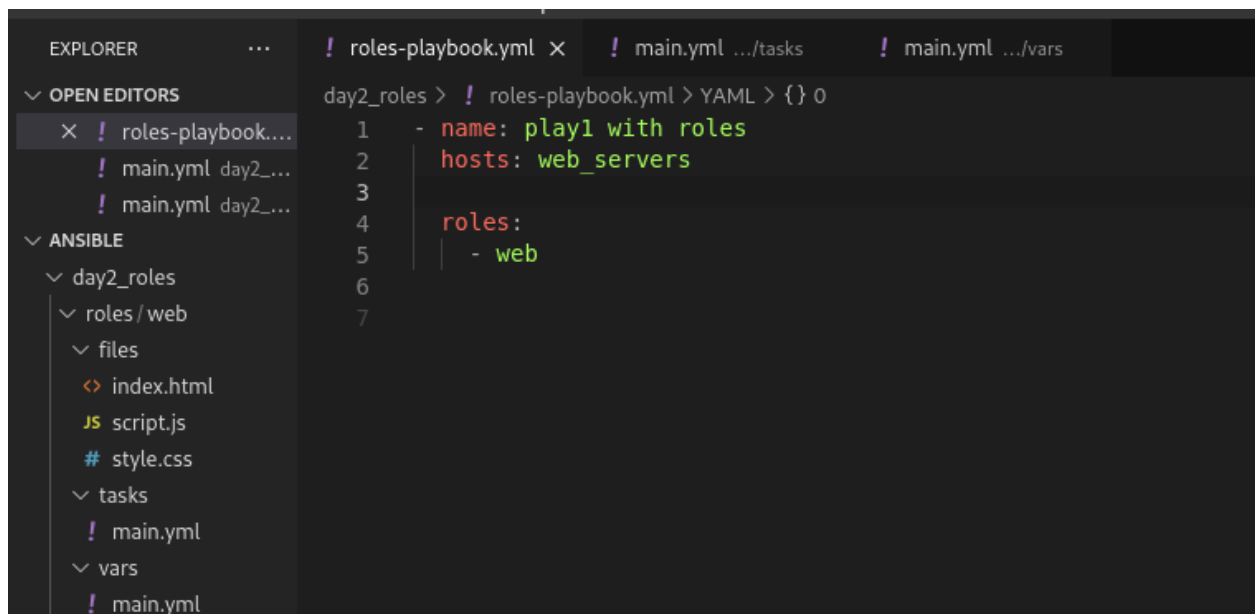
TASK [task1 (Copy dynamic content from controller to host 1 using templates)] *****
changed: [172.17.0.2]

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

• [Sun May 07] @samy: $curl 172.17.0.2
• [Sun May 07] @samy: $curl 172.17.0.2
o ===== Hello! this is Ansible course =====[Sun May 07] @samy: $
```

- ▶ Create your first role with name (web)
- ▶ The task book will include:
  1. installing a package  
(get the package name from vars)
  2. 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)

## ROLES



The screenshot shows the Visual Studio Code interface with an Ansible project. The Explorer sidebar on the left displays the project structure under the 'ANSIBLE' section:

- day2\_roles
  - roles/web
    - files
      - index.html
      - script.js
      - style.css
    - tasks
      - main.yml
    - vars
      - main.yml

The main editor area shows the 'roles-playbook.yml' file with the following YAML content:

```
1 - name: play1 with roles
2   hosts: web_servers
3
4   roles:
5     - web
```

EXPLORER

OPEN EDITORS

- roles-playbook....
- main.yml day2\_...
- main.yml day2\_...

ANSIBLE

- day2\_roles
  - roles/web
    - files
      - index.html
      - script.js
      - style.css
    - tasks
      - main.yml
    - vars

day2\_roles > roles > web > tasks > ! main.yml > ...

Ansible Tasks Schema - Ansible tasks file (ansible.json)

```
1 - name: task1 (installing a package)
2   apt:
3     name: "{{ package_name }}"
4     state: latest
5
```

EXPLORER

OPEN EDITORS

- roles-playbook....
- main.yml day2\_...
- main.yml day2\_...

ANSIBLE

- day2\_roles
  - roles/web
    - files
      - index.html
      - script.js

day2\_roles > roles > web > vars > ! main.yml > ...

Ansible Vars File - Ansible variables File (vars.json)

```
1 package_name: nginx
2
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
[Sun May 07] @samy: $ansible-playbook ./roles-playbook.yml
BECOME password:

PLAY [play1 with roles] *****

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

TASK [web : task1 (installing a package)] *****
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
```

EXPLORER...roles-playbook.ymlmain.yml .../tasks ×main.yml .../vars

OPEN EDITORSroles-playbook...main.yml day2...main.yml day2...ANSIBLEday2\_rolesroles/webfilesindex.htmlscript.jssstyle.csstasksmain.ymlvarsmain.ymlansible.cfginventoryroles-playbook.ymlansible.cfgconditions.yml

day2\_roles > roles > web > tasks > ! main.yml > Google Cloud Code > whenAnsible Tasks Schema - Ansible tasks file (ansible.json)  
1 - name: task1 (installing a package)  
2 apt:  
3 name: "{{ package\_name }}"  
4 state: latest  
5 register: install\_result  
6  
7 - name: task2 (copying)  
8 copy:  
9 src: "{{ item }}"  
10 dest: /var/www/html/  
11 loop: "{{ my\_files }}"  
12 when: install\_result.changed == true  
13 register: copy\_result  
14  
15 - name: task3 (Restart)  
16 sysvinit:  
17 name: "{{ package\_name }}"  
18 state: restarted  
19 enabled: yes  
20 when: copy\_result.changed == true  
21

EXPLORER...roles-playbook.ymlmain.yml .../tasksmain.yml .../vars ×

OPEN EDITORSroles-playbook...main.yml day2...main.yml day2...ANSIBLEday2\_rolesroles/webfiles

day2\_roles > roles > web > vars > ! main.yml > Google Cloud Code > [ ] my\_filesAnsible Vars File - Ansible variables File (vars.json)  
1 package\_name: nginx  
2  
3 my\_files:  
4 - index.html  
5 - style.css  
6 - script.js  
7

PROBLEMSOUTPUTDEBUG CONSOLETERMINAL

• [Sun May 07] @samy: \$ansible-playbook ./roles-playbook.yml  
BECOME password:

PLAY [play1 with roles] \*\*\*\*\*

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

TASK [web : task1 (installing a package)] \*\*\*\*\*  
changed: [172.17.0.2]

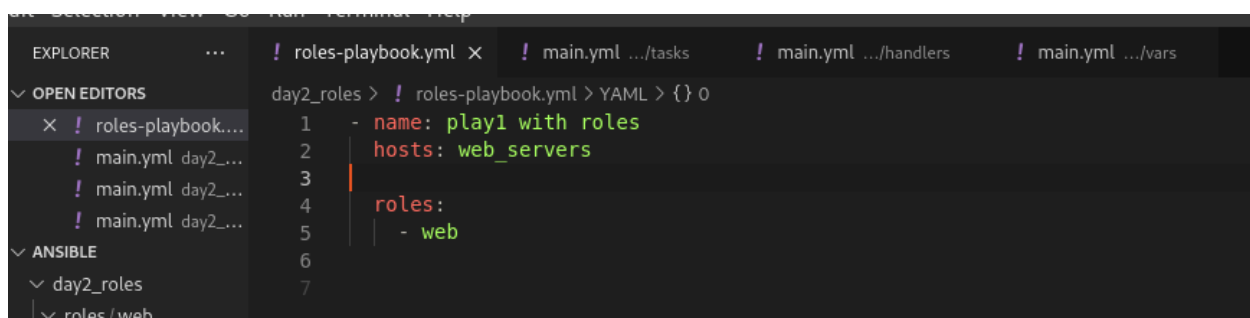
TASK [web : task2 (copying)] \*\*\*\*\*  
changed: [172.17.0.2] => (item=index.html)  
changed: [172.17.0.2] => (item=style.css)  
changed: [172.17.0.2] => (item=script.js)

TASK [web : task3 (Restart)] \*\*\*\*\*  
changed: [172.17.0.2]

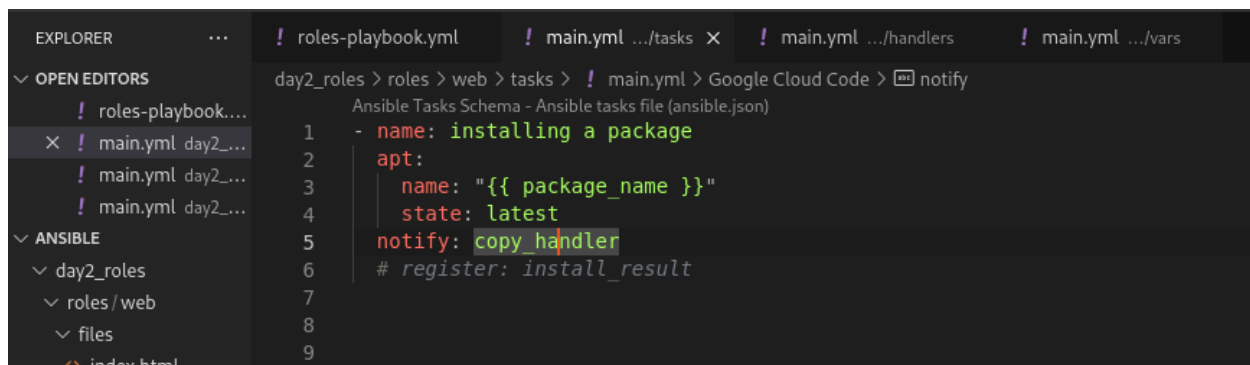
PLAY RECAP \*\*\*\*\*  
172.17.0.2 : ok=4 changed=3 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

- ▶ Create your first role with name (web)
- ▶ The task book will include:
  1. installing a package  
(get the package name from vars)
  2. 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 using Handlers)
- ▶ Restart the service of the installed package  
(will be executed using Handlers chaining)

# HANDLERS



```
! roles-playbook.yml x ! main.yml .../tasks ! main.yml .../handlers ! main.yml .../vars
day2_roles > ! roles-playbook.yml > YAML > {} 0
1 - name: play1 with roles
2 hosts: web_servers
3
4 roles:
5 - web
6
7
```



```
! roles-playbook.yml ! main.yml .../tasks x ! main.yml .../handlers ! main.yml .../vars
day2_roles > roles > web > tasks > ! main.yml > Google Cloud Code > notify
Ansible Tasks Schema - Ansible tasks file (ansible.json)
1 - name: installing a package
2 apt:
3 name: "{{ package_name }}"
4 state: latest
5 notify: copy_handler
6 # register: install_result
7
8
9
```

The screenshot shows the VS Code editor with the 'roles-playbook.yml' file open. The Explorer sidebar on the left shows the project structure: 'roles-playbook.yml', 'main.yml', 'day2\_roles', 'roles/web', 'files', 'index.html', 'script.js', 'style.css', 'handlers', 'main.yml', 'tasks', 'main.yml', 'vars', 'main.yml'. The main editor area shows the 'copy\_handler' task in the 'handlers' section of the 'roles/web' directory. The task is defined as follows:

```
1 - name: copy_handler
2   copy:
3     src: "{{ item }}"
4     dest: /var/www/html/
5   loop: "{{ my_files }}"
6   changed_when: true
7   notify: restart_handler
8
9 - name: restart_handler
10  sysvinit:
11    name: "{{ package_name }}"
12    state: restarted
13    enabled: yes
14
```

The screenshot shows the VS Code editor with the 'main.yml' file open. The Explorer sidebar on the left shows the project structure: 'roles-playbook.yml', 'main.yml', 'day2\_roles', 'roles/web', 'vars', 'main.yml'. The main editor area shows the 'my\_files' variable in the 'vars' section of the 'roles/web' directory. The variable is defined as follows:

```
1 package_name: nginx
2
3 my_files:
4   - index.html
5   - style.css
6   - script.js
7
```

The screenshot shows a terminal window with the output of an Ansible playbook run. The output is as follows:

```
• [Sun May 07] @samy: $ansible-playbook ./roles-playbook.yml
BECOME password:

PLAY [play1 with roles] *****

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

TASK [web : installing a package] *****
changed: [172.17.0.2]

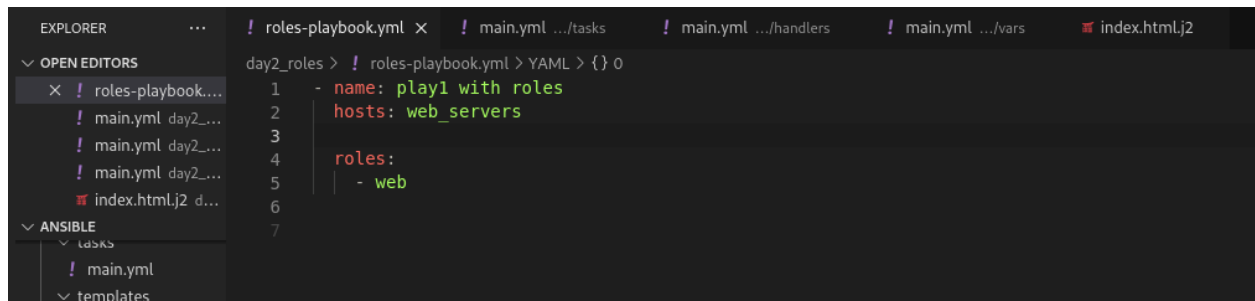
RUNNING HANDLER [web : copy_handler] *****
changed: [172.17.0.2] => (item=index.html)
changed: [172.17.0.2] => (item=style.css)
changed: [172.17.0.2] => (item=script.js)

RUNNING HANDLER [web : restart_handler] *****
changed: [172.17.0.2]

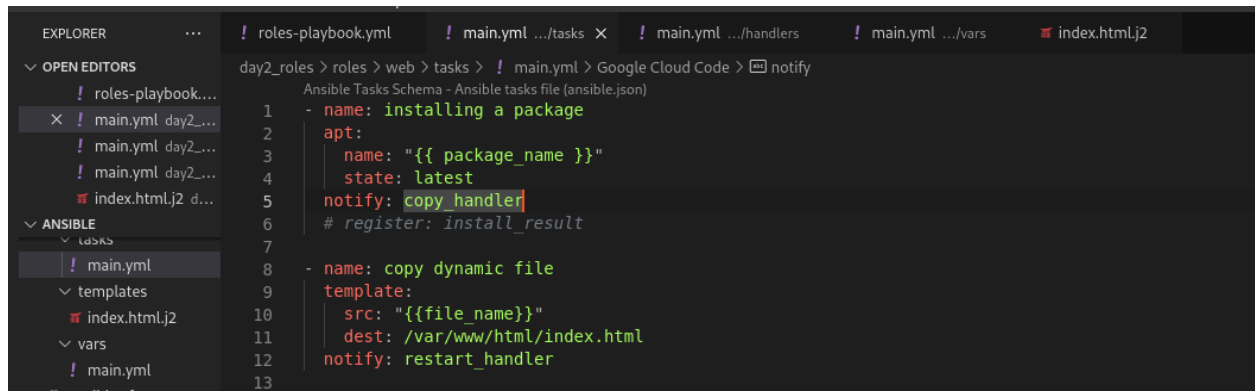
PLAY RECAP *****
172.17.0.2 : ok=4 changed=3 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
```

- ▶ Create your first role with name (web)
- ▶ The task book will include:
  1. installing a package  
(get the package name from vars)
  2. Copying a file from controller to host using template  
(get the template name & template message from vars)  
(the actual template file will be stored in ./roles/web/templates)  
(will also notify the restart handler)
  3. 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 using Handlers)
- ▶ Restart the service of the installed package  
(will be executed using Handlers chaining)

# TEMPLATES



```
EXPLORER  ...  ! roles-playbook.yml x  ! main.yml .../tasks  ! main.yml .../handlers  ! main.yml .../vars  index.htmlj2
└ OPEN EDITORS
  x ! roles-playbook...
  ! main.yml day2_...
  ! main.yml day2_...
  ! main.yml day2_...
  ! main.yml day2_...
  index.htmlj2 d...
└ ANSIBLE
  tasks
  ! main.yml
  templates
day2_roles > ! roles-playbook.yml > YAML > {} 0
1 - name: play1 with roles
2   hosts: web_servers
3
4   roles:
5     - web
6
7
```



```
EXPLORER  ...  ! roles-playbook.yml  ! main.yml .../tasks x  ! main.yml .../handlers  ! main.yml .../vars  index.htmlj2
└ OPEN EDITORS
  ! roles-playbook...
  x ! main.yml day2_...
  ! main.yml day2_...
  ! main.yml day2_...
  ! main.yml day2_...
  index.htmlj2 d...
└ ANSIBLE
  tasks
  ! main.yml
  templates
  vars
  ! main.yml
  ! ansible.cfg
day2_roles > roles > web > tasks > ! main.yml > Google Cloud Code > notify
Ansible Tasks Schema - Ansible tasks file (ansible.json)
1 - name: installing a package
2   apt:
3     name: "{{ package_name }}"
4     state: latest
5     notify: copy_handler
6     # register: install_result
7
8 - name: copy dynamic file
9   template:
10     src: "{{ file_name }}"
11     dest: /var/www/html/index.html
12     notify: restart_handler
13
```

```
day2_roles > roles > web > handlers > ! main.yml > Google Cloud Code > name
Ansible Tasks Schema - Ansible tasks file (ansible.json)
1  - name: copy_handler
2    copy:
3      src: "{{ item }}"
4      dest: /var/www/html/
5      loop: "{{ my_files }}"
6      changed_when: true
7      notify: restart_handler
8
9  - name: restart_handler
10    sysvinit:
11      name: "{{ package_name }}"
12      state: restarted
13      enabled: yes
14
```

```
day2_roles > roles > web > templates > index.htmlj2
1  {{ my_message }}
```

```
day2_roles > roles > web > vars > ! main.yml > Google Cloud Code > [ ] my_files
Ansible Vars File - Ansible variables File (vars.json)
1  package_name: nginx
2
3  file_name: index.htmlj2
4  my_message: Hi, this is template test!
5
6  my_files:
7    # - index.html
8    - style.css
9    - script.js
10
```

```
[Sun May 07] @samy: $ansible-playbook ./roles-playbook.yml
BECOME password:

PLAY [play1 with roles] *****

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

TASK [web : installing a package] *****
changed: [172.17.0.2]

TASK [web : copy dynamic file] *****
changed: [172.17.0.2]

RUNNING HANDLER [web : copy_handler] *****
changed: [172.17.0.2] => (item=style.css)
changed: [172.17.0.2] => (item=script.js)

RUNNING HANDLER [web : restart_handler] *****
changed: [172.17.0.2]

PLAY RECAP *****
172.17.0.2 : ok=5  changed=4  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0

[Sun May 07] @samy: $curl 172.17.0.2
[Sun May 07] @samy: $curl 172.17.0.2
[Sun May 07] @samy: $curl 172.17.0.2
.. Hi, this is template test![Sun May 07] @samy: $
```