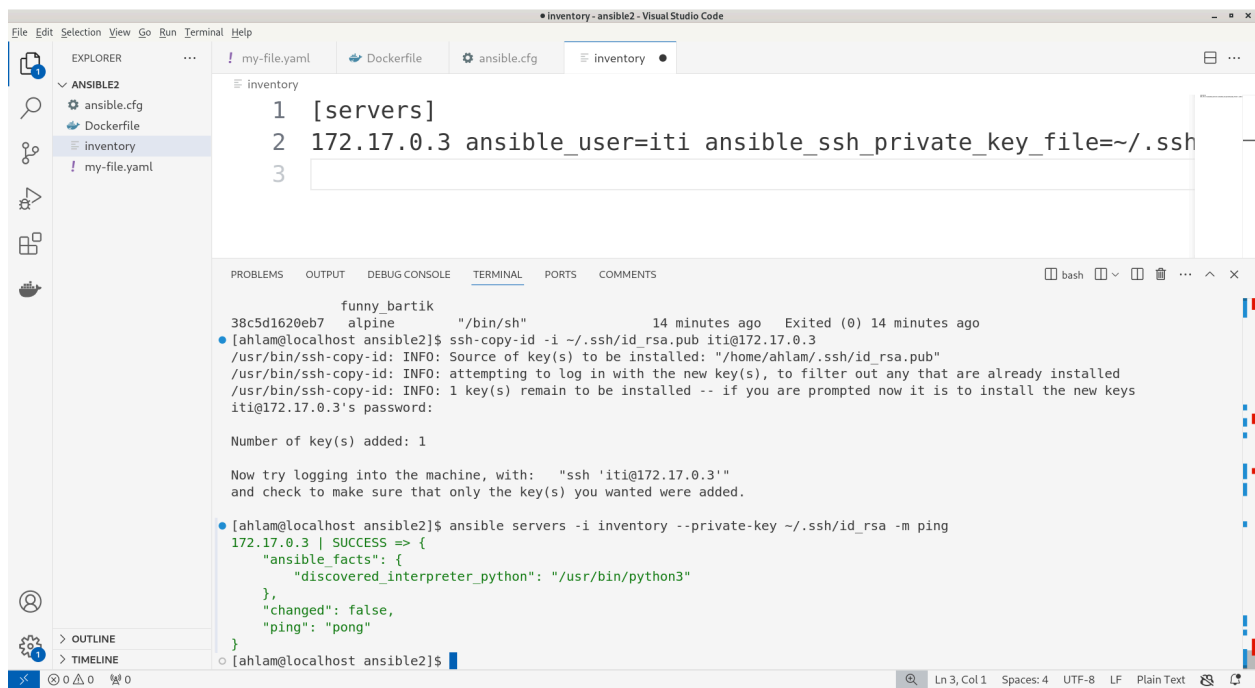


Write simple playbook file Add two tasks (apt update – apt install nginx) Add tags to first task: update Add tags to second task: install Run only the (apt update) task Example: ansible-playbook my-playbook.yml --tags update Add one task with “tags: always” and run the previous command again



The screenshot shows the Visual Studio Code interface with the 'inventory' file open in the editor. The inventory file contains the following content:

```
1 [servers]
2 172.17.0.3 ansible_user=iti ansible_ssh_private_key_file=~/.ssh
3
```

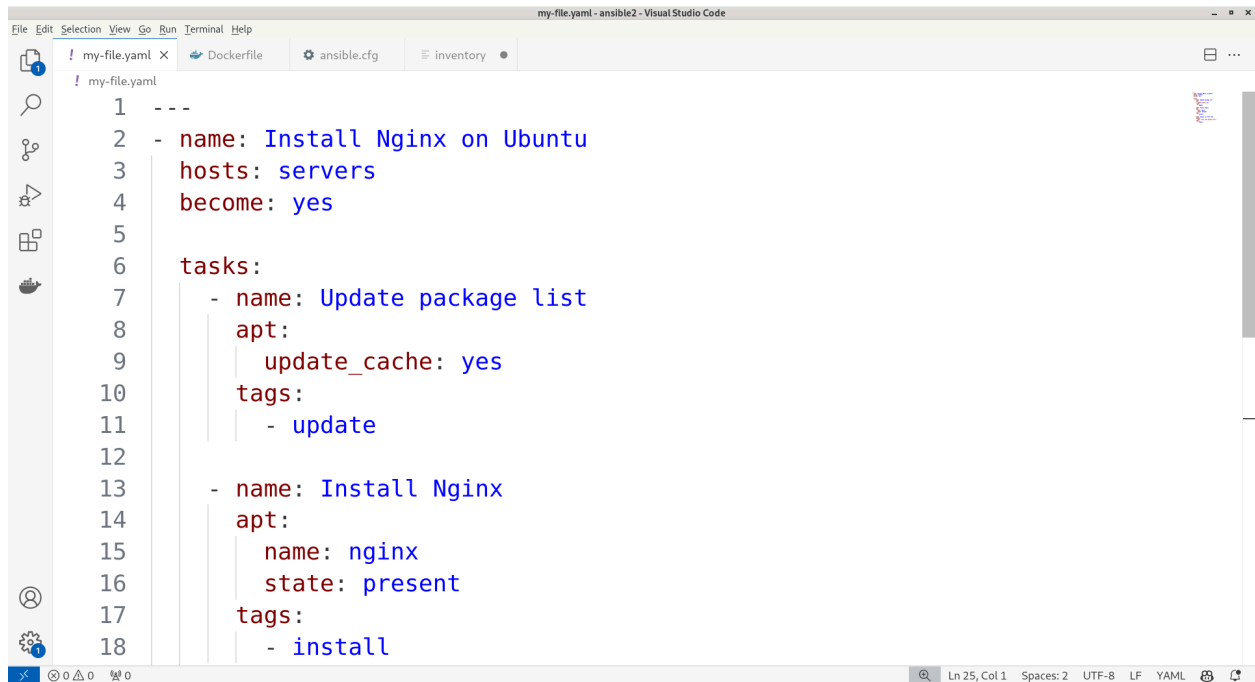
The terminal window shows the output of the following commands:

```
funny_bartik
38c5d1620eb7 alpine "/bin/sh" 14 minutes ago Exited (0) 14 minutes ago
[ahlam@localhost ansible2]$ ssh-copy-id -i ~/.ssh/id_rsa.pub iti@172.17.0.3
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/ahlam/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
iti@172.17.0.3's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'iti@172.17.0.3'"
and check to make sure that only the key(s) you wanted were added.

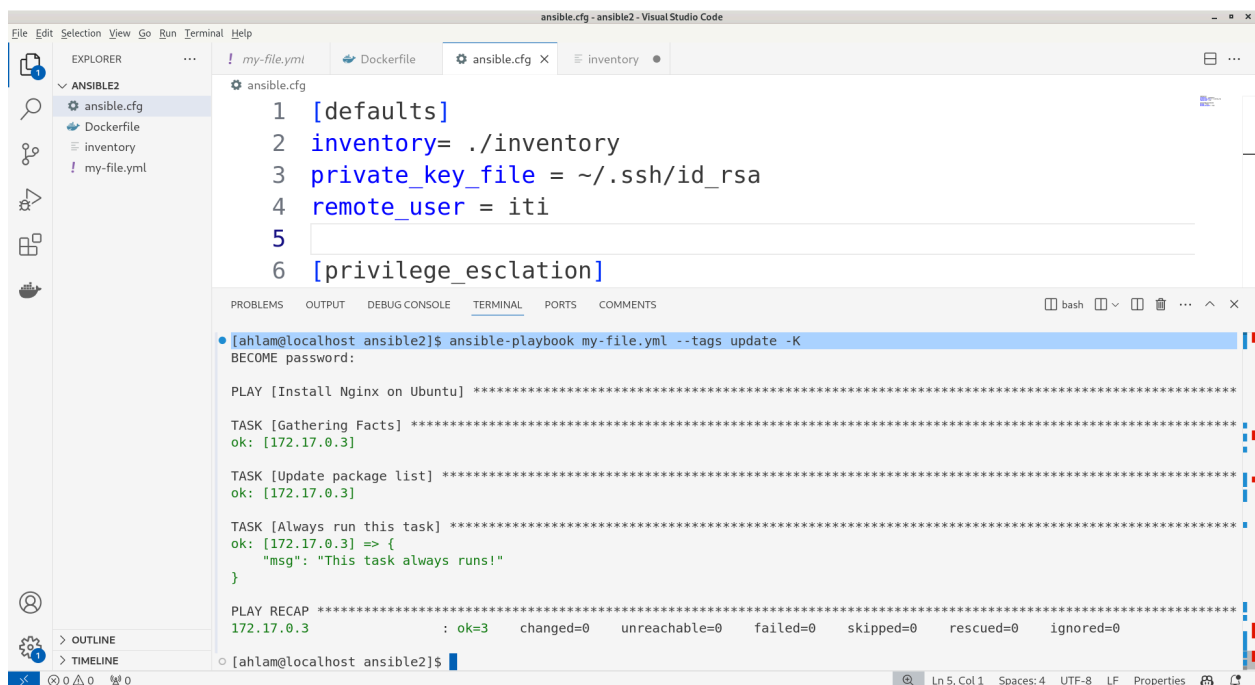
[ahlam@localhost ansible2]$ ansible servers -i inventory --private-key ~/.ssh/id_rsa -m ping
172.17.0.3 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
[ahlam@localhost ansible2]$
```



The screenshot shows the Visual Studio Code editor with a file named 'my-file.yml' open. The file contains an Ansible playbook with the following content:

```
1 ---
2 - name: Install Nginx on Ubuntu
3   hosts: servers
4   become: yes
5
6   tasks:
7     - name: Update package list
8       apt:
9         update_cache: yes
10      tags:
11        - update
12
13     - name: Install Nginx
14       apt:
15         name: nginx
16         state: present
17      tags:
18        - install
```

## Update tag



The screenshot shows the Visual Studio Code editor with the 'ansible.cfg' file open. The file contains the following configuration:

```
1 [defaults]
2 inventory= ./inventory
3 private_key_file = ~/.ssh/id_rsa
4 remote_user = iti
5
6 [privilege_escalation]
```

Below the editor, the terminal window shows the execution of the Ansible playbook with the 'update' tag. The output is as follows:

```
[ahlam@localhost ansible2]$ ansible-playbook my-file.yml --tags update -K
BECOME password:

PLAY [Install Nginx on Ubuntu] *****

TASK [Gathering Facts] *****
ok: [172.17.0.3]

TASK [Update package list] *****
ok: [172.17.0.3]

TASK [Always run this task] *****
ok: [172.17.0.3] => {
  "msg": "This task always runs!"
}

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

[ahlam@localhost ansible2]$
```

## Install nginx tag

```
2 - name: Install Nginx on Ubuntu
6 tasks:

13   - name: Install Nginx
14     apt:
15       name: nginx
16       state: present
```

```
[ahlam@localhost ansible2]$ ansible-playbook my-file.yml --tags install -K
BECOME password:

PLAY [Install Nginx on Ubuntu] *****

TASK [Gathering Facts] *****
ok: [172.17.0.3]

TASK [Install Nginx] *****
changed: [172.17.0.3]

TASK [Always run this task] *****
ok: [172.17.0.3] => {
  "msg": "This task always runs!"
}

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

[ahlam@localhost ansible2]$
```

## All tasks

```
1 ---
2 - name: Install Nginx on Ubuntu
3   hosts: servers
4   become: yes
```

```
[ahlam@localhost ansible2]$ ansible-playbook my-file.yml -K
BECOME password:

PLAY [Install Nginx on Ubuntu] *****

TASK [Gathering Facts] *****
ok: [172.17.0.3]

TASK [Update package list] *****
ok: [172.17.0.3]

TASK [Install Nginx] *****
ok: [172.17.0.3]

TASK [Always run this task] *****
ok: [172.17.0.3] => {
  "msg": "This task always runs!"
}

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

[ahlam@localhost ansible2]$
```

Define these variables (package\_name,  
package\_version)

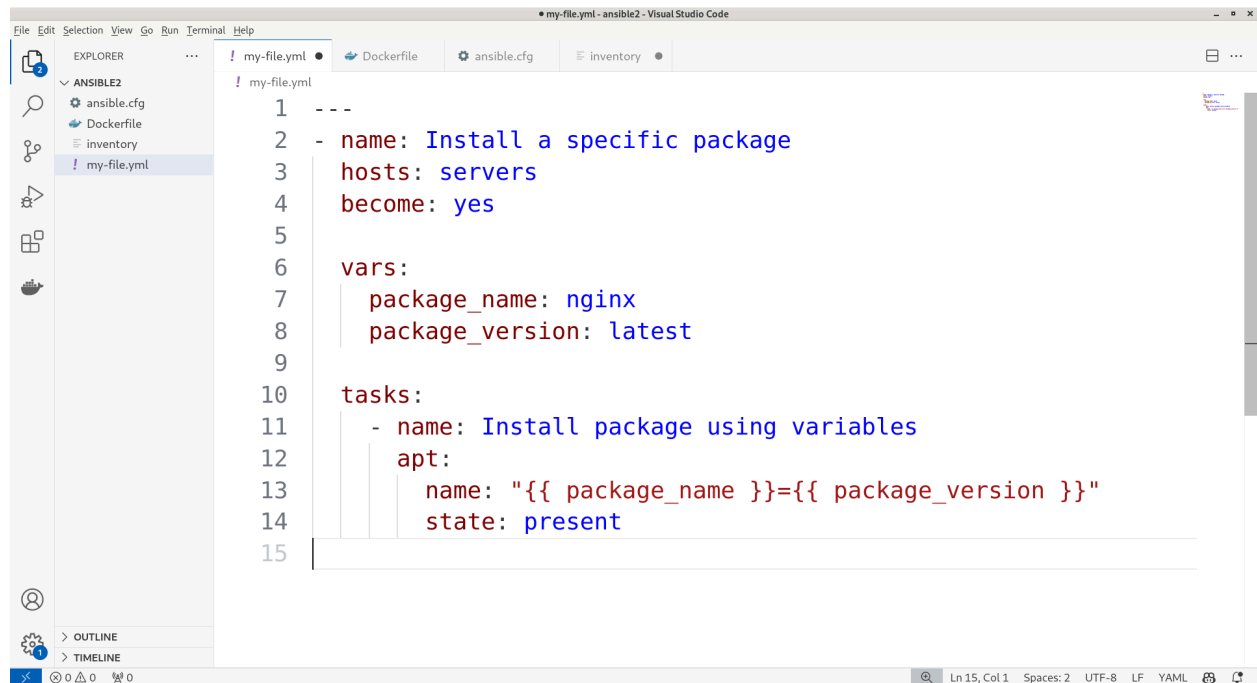
on playbook level

on inventory level

on command line level

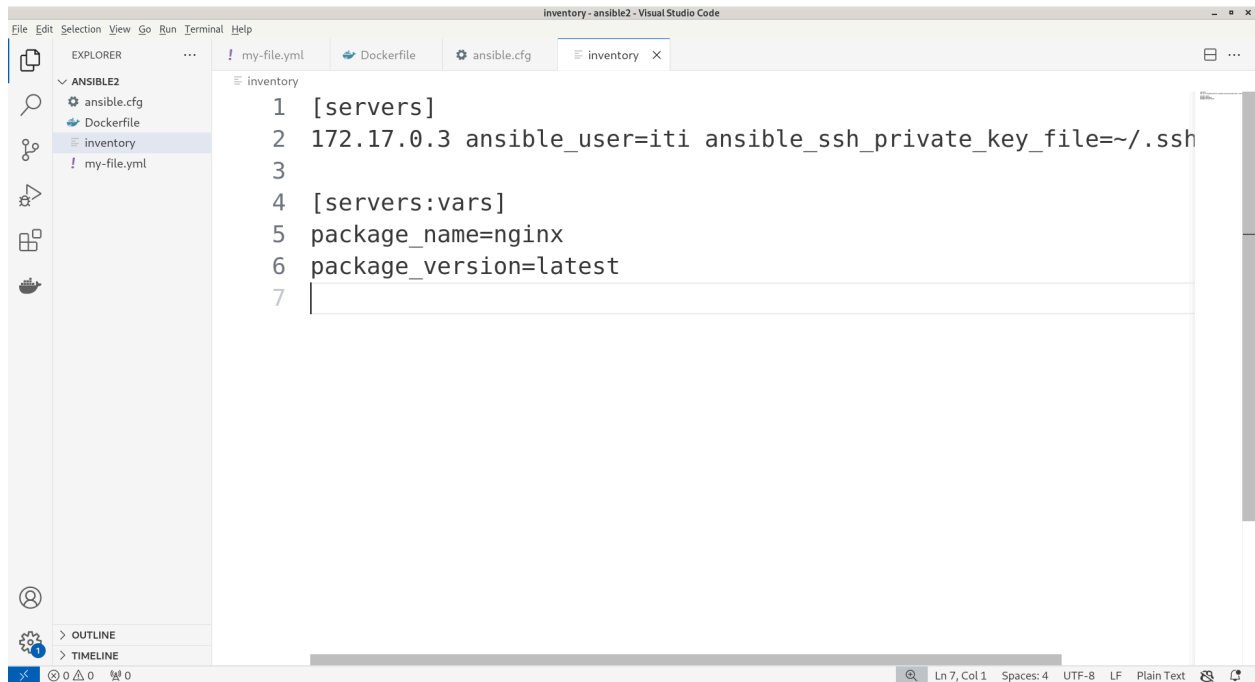
Use apt module with the package name and version  
from your variables

Vars in playbook level



```
1 ---
2 - name: Install a specific package
3   hosts: servers
4   become: yes
5
6   vars:
7     package_name: nginx
8     package_version: latest
9
10  tasks:
11    - name: Install package using variables
12      apt:
13        name: "{{ package_name }}={{ package_version }}"
14        state: present
15
```

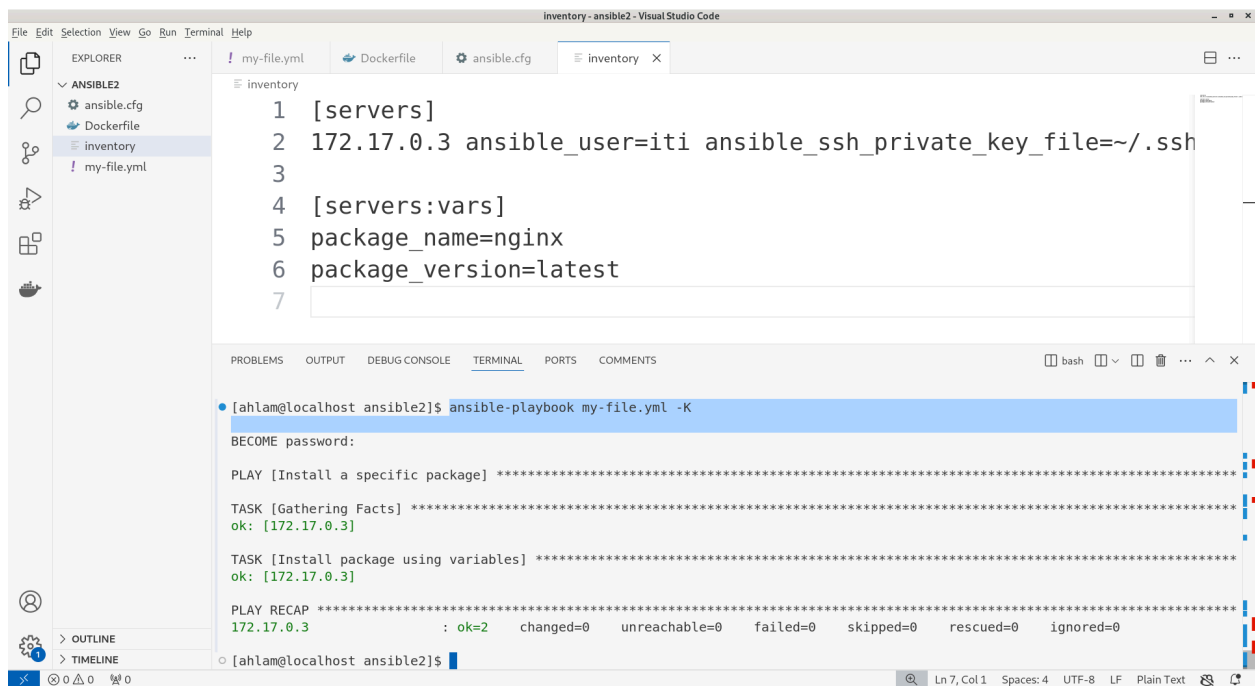
Vars on inventory level



The screenshot shows the Visual Studio Code editor with the 'inventory' file open. The file content is as follows:

```
1 [servers]
2 172.17.0.3 ansible_user=iti ansible_ssh_private_key_file=~/.ssh
3
4 [servers:vars]
5 package_name=nginx
6 package_version=latest
7
```

Took vars from inventory level



The screenshot shows the Visual Studio Code editor with the 'inventory' file open, and the terminal window displaying the output of an Ansible playbook execution. The terminal output is as follows:

```
[ahlam@localhost ansible2]$ ansible-playbook my-file.yml -K
BECOME password:

PLAY [Install a specific package] *****

TASK [Gathering Facts] *****
ok: [172.17.0.3]

TASK [Install package using variables] *****
ok: [172.17.0.3]

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

[ahlam@localhost ansible2]$
```

Override vars using cmd

The screenshot shows the Visual Studio Code interface with a terminal window displaying the output of an Ansible playbook. The terminal output is as follows:

```
my-file.yml - ansible2 - Visual Studio Code
my-file.yml
2 - name: Install a specific package

[ahlam@localhost ansible2]$ ansible-playbook my-file.yml -e "package_name=apache2 package_version=2.4.41-4ubuntu3.14" -K
BECOME password:

PLAY [Install a specific package] *****

TASK [Gathering Facts] *****
ok: [172.17.0.3]

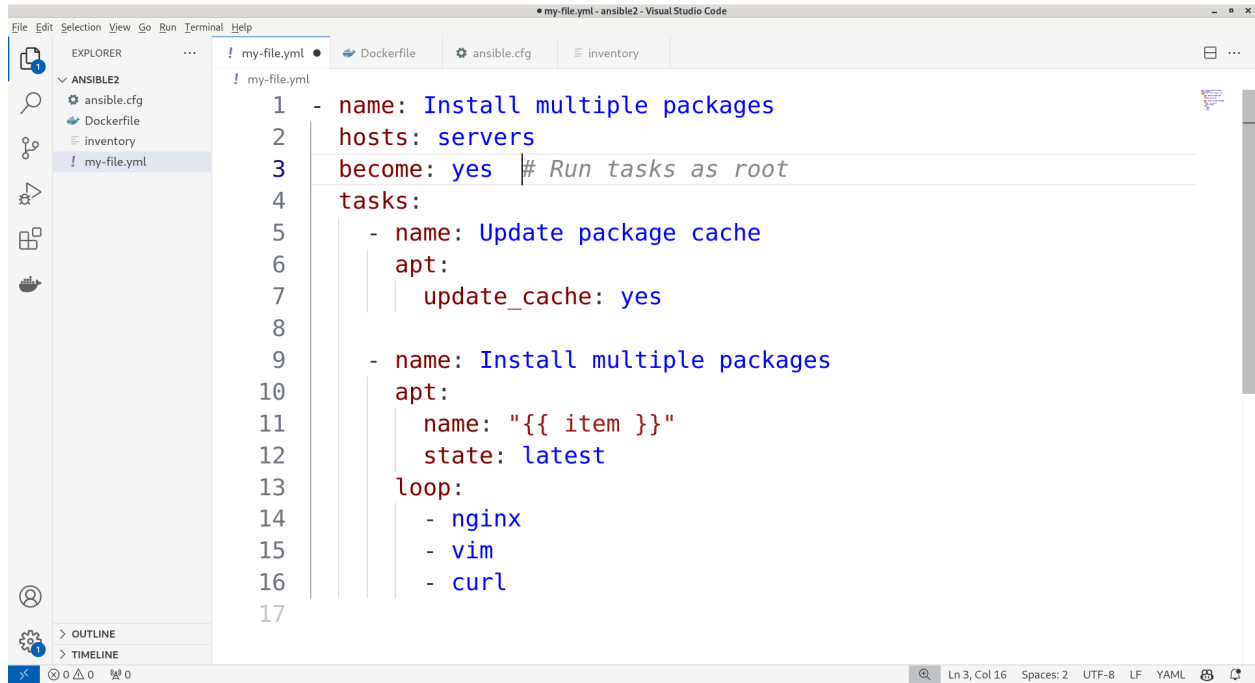
TASK [Install package using variables] *****
changed: [172.17.0.3]

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

[ahlam@localhost ansible2]$
```

Installed apache instead of nginx

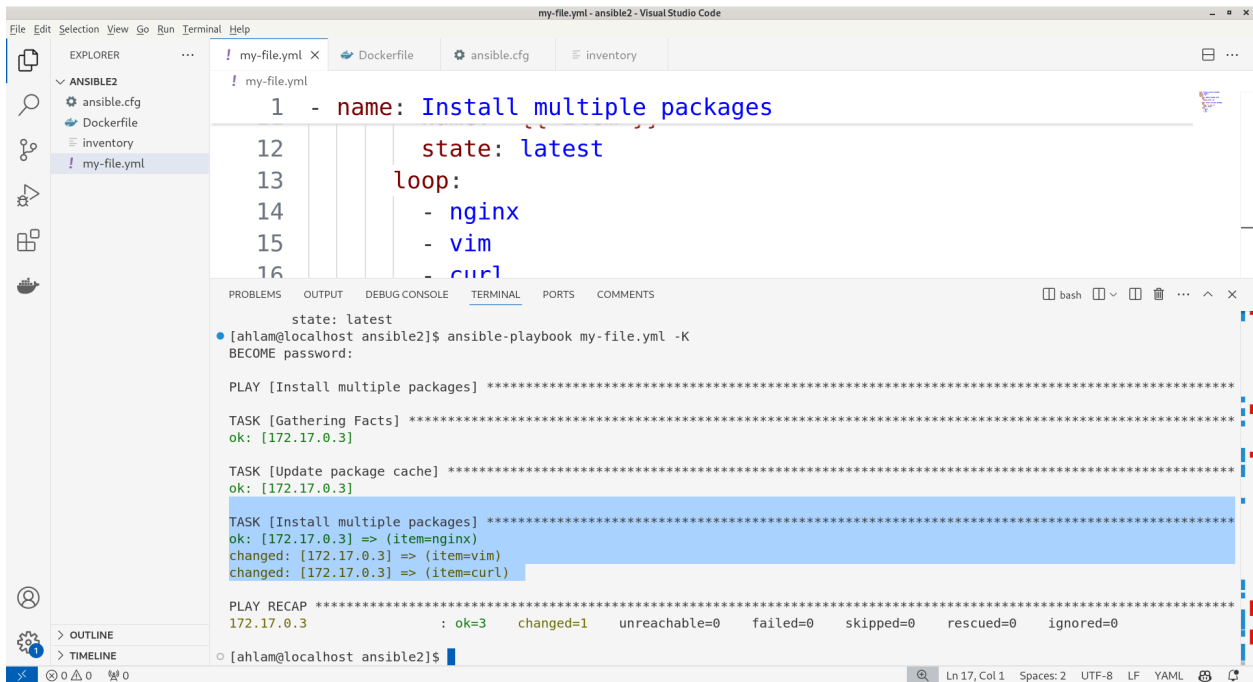
# Loop over a list of packages and install latest versions.



The screenshot shows the Visual Studio Code interface with a file named 'my-file.yml' open. The file contains an Ansible playbook with the following tasks:

```
1 - name: Install multiple packages
2   hosts: servers
3   become: yes # Run tasks as root
4   tasks:
5     - name: Update package cache
6       apt:
7         update_cache: yes
8
9     - name: Install multiple packages
10      apt:
11        name: "{{ item }}"
12        state: latest
13      loop:
14        - nginx
15        - vim
16        - curl
17
```

The interface includes a sidebar with the Explorer view showing the file structure, and a bottom status bar indicating the current line and column (Ln 3, Col 16).



```
1 - name: Install multiple packages
12   state: latest
13   loop:
14     - nginx
15     - vim
16     - curl

state: latest
[ahlam@localhost ansible2]$ ansible-playbook my-file.yml -K
BECOME password:

PLAY [Install multiple packages] *****

TASK [Gathering Facts] *****
ok: [172.17.0.3]

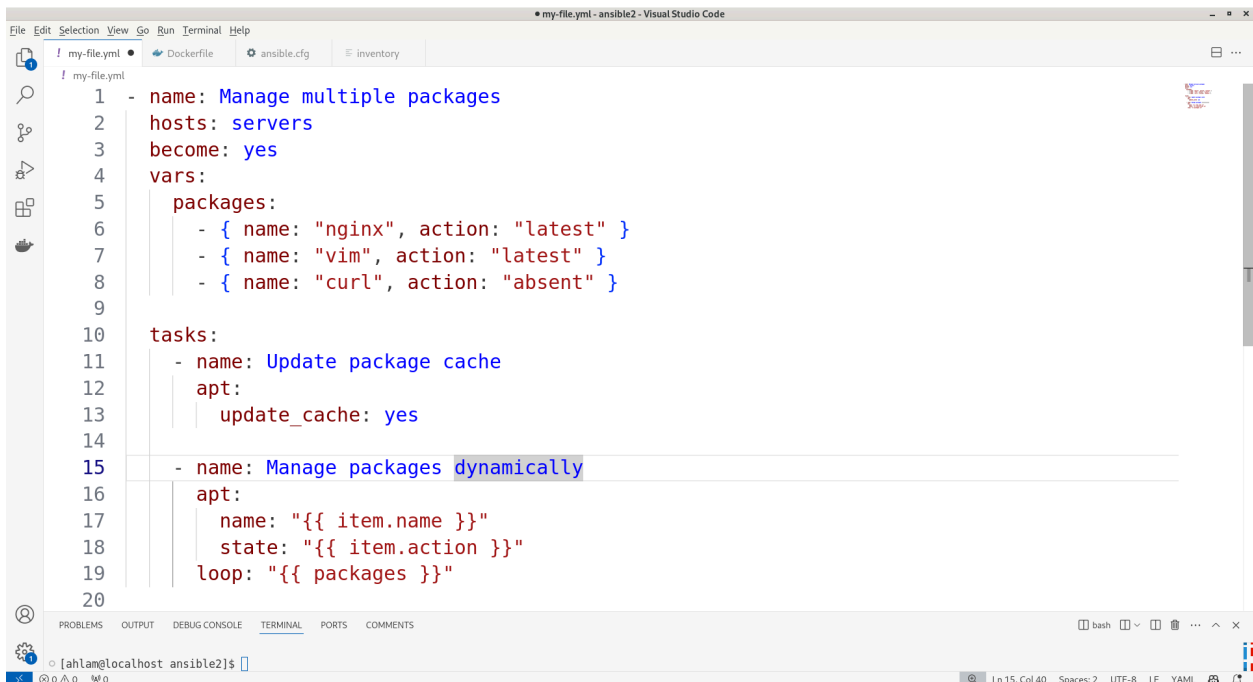
TASK [Update package cache] *****
ok: [172.17.0.3]

TASK [Install multiple packages] *****
ok: [172.17.0.3] => (item=nginx)
changed: [172.17.0.3] => (item=vim)
changed: [172.17.0.3] => (item=curl)

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

[ahlam@localhost ansible2]$
```

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



```
1 - name: Manage multiple packages
2   hosts: servers
3   become: yes
4   vars:
5     packages:
6       - { name: "nginx", action: "latest" }
7       - { name: "vim", action: "latest" }
8       - { name: "curl", action: "absent" }
9
10  tasks:
11    - name: Update package cache
12      apt:
13        update_cache: yes
14
15    - name: Manage packages dynamically
16      apt:
17        name: "{{ item.name }}"
18        state: "{{ item.action }}"
19        loop: "{{ packages }}"
20
```



The image shows a Visual Studio Code editor window with the title bar "my-file.yml - ansible2 - VisualStudio Code". The editor has a sidebar on the left with icons for Explorer, Search, Source Control, Run and Debug, and Extensions. The main editor area displays a file named "my-file.yml" with the following content:

```
1 - name: Manage multiple packages
2   hosts: servers
3   become: yes
4   vars:
5     packages:
```

Below the editor is a terminal window with the title bar "bash". The terminal shows the execution of the Ansible playbook:

```
[ahlam@localhost ansible2]$ ansible-playbook my-file.yml -K
BECOME password:

PLAY [Manage multiple packages] *****

TASK [Gathering Facts] *****
ok: [172.17.0.3]

TASK [Update package cache] *****
ok: [172.17.0.3]

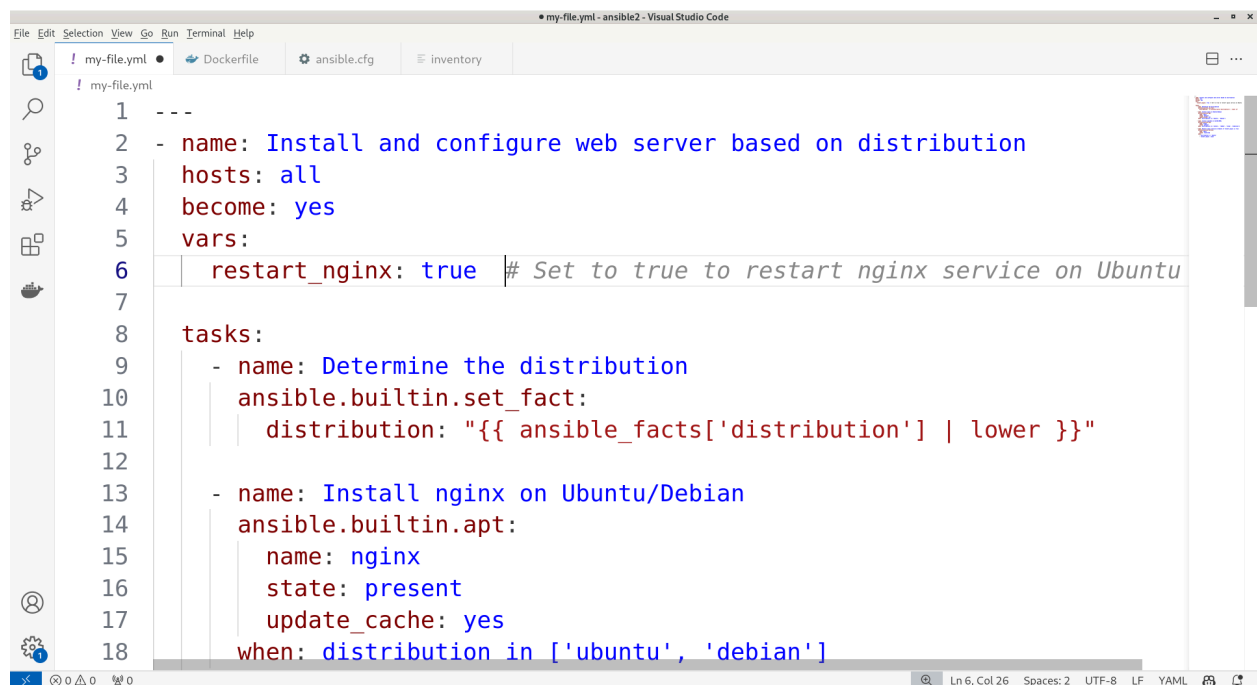
TASK [Manage packages dynamically] *****
ok: [172.17.0.3] => (item={'name': 'nginx', 'action': 'latest'})
ok: [172.17.0.3] => (item={'name': 'vim', 'action': 'latest'})
changed: [172.17.0.3] => (item={'name': 'curl', 'action': 'absent'})

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

[ahlam@localhost ansible2]$
```

The terminal output is displayed in a light blue background. The status bar at the bottom of the editor shows "Ln 15, Col 40", "Spaces: 2", "UTF-8", "LF", and "YAML".

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



```
1 ---
2 - name: Install and configure web server based on distribution
3   hosts: all
4   become: yes
5   vars:
6     restart_nginx: true # Set to true to restart nginx service on Ubuntu
7
8   tasks:
9     - name: Determine the distribution
10       ansible.builtin.set_fact:
11         distribution: "{{ ansible_facts['distribution'] | lower }}"
12
13     - name: Install nginx on Ubuntu/Debian
14       ansible.builtin.apt:
15         name: nginx
16         state: present
17         update_cache: yes
18       when: distribution in ['ubuntu', 'debian']
```

my-file.yml - ansible2 - Visual Studio Code

File Edit Selection View Go Run Terminal Help

! my-file.yml x Dockerfile ansible.cfg inventory

! my-file.yml

2 - name: Install and configure web server based on distribution

23 state: present

24 when: distribution in ['centos', 'redhat', 'rocky', 'almalinux']

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

bash

- [ahlam@localhost ansible2]\$ ansible-playbook my-file.yml -K

BECOME password:

PLAY [Install and configure web server based on distribution] \*\*\*\*\*

TASK [Gathering Facts] \*\*\*\*\*

ok: [172.17.0.3]

TASK [Determine the distribution] \*\*\*\*\*

ok: [172.17.0.3]

TASK [Install nginx on Ubuntu/Debian] \*\*\*\*\*

ok: [172.17.0.3]

TASK [Install apache2 on CentOS/RHEL] \*\*\*\*\*

skipping: [172.17.0.3]

TASK [Restart nginx service on Ubuntu if restart\_nginx is true] \*\*\*\*\*

[WARNING]: Target is a chroot or systemd is offline. This can lead to false positives or prevent the init system tools from working.

ok: [172.17.0.3]

PLAY RECAP \*\*\*\*\*

172.17.0.3 : ok=4 changed=0 unreachable=0 failed=0 skipped=1 rescued=0 ignored=0

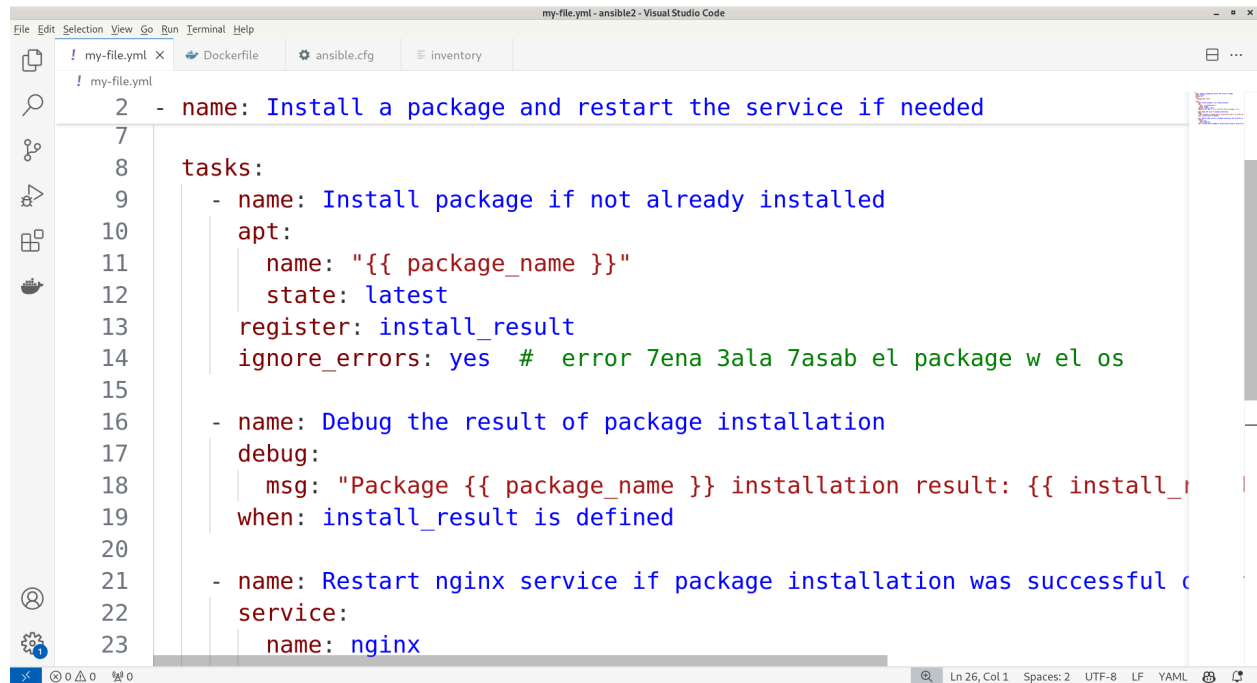
[ahlam@localhost ansible2]\$

Ln 34, Col 9 Spaces: 2 UTF-8 LF YAML

# REGISTER & WHEN

View the value of your register variable using debug module

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

A screenshot of the Visual Studio Code editor interface. The title bar at the top reads 'my-file.yml - ansible2 - Visual Studio Code'. The editor has several tabs open: 'my-file.yml', 'Dockerfile', 'ansible.cfg', and 'inventory'. The 'my-file.yml' tab is active, showing an Ansible playbook. The left sidebar contains icons for Explorer, Search, Source Control, Run and Debug, Extensions, and Settings. The main editor area displays the following YAML code:

```
2 - name: Install a package and restart the service if needed
7
8 tasks:
9   - name: Install package if not already installed
10     apt:
11       name: "{{ package_name }}"
12       state: latest
13       register: install_result
14       ignore_errors: yes # error 7ena 3ala 7asab el package w el os
15
16   - name: Debug the result of package installation
17     debug:
18       msg: "Package {{ package_name }} installation result: {{ install_result }}"
19       when: install_result is defined
20
21   - name: Restart nginx service if package installation was successful
22     service:
23       name: nginx
```

The status bar at the bottom indicates 'Ln 26, Col 1', 'Spaces: 2', 'UTF-8', 'LF', and 'YAML'.

The image shows a Visual Studio Code editor window titled "my-file.yml - ansible2 - Visual Studio Code". The editor has several tabs: "my-file.yml", "Dockerfile", "ansible.cfg", and "inventory". The "my-file.yml" tab is active, displaying an Ansible playbook with the following content:

```
2 - name: Install a package and restart the service if needed
8 tasks:
18     msg: "Package {{ package_name }} installation result: {{ install_result }}"
19     when: install_result is defined
20
21 - name: Restart nginx service if package installation was successful or changed
22   service:
```

Below the editor, the "TERMINAL" panel is open, showing the output of the Ansible playbook execution. The output is as follows:

```
TASK [Gathering Facts] *****
ok: [172.17.0.3]

TASK [Install package if not already installed] *****
ok: [172.17.0.3]

TASK [Debug the result of package installation] *****
ok: [172.17.0.3] => {
  "msg": "Package nginx installation result: {'changed': False, 'cache_updated': False, 'cache_update_time': 1738589312, 'failed': False}"
}

TASK [Restart nginx service if package installation was successful or changed] *****
changed: [172.17.0.3]

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

[ahlam@localhost ansible2]$
```

The terminal output indicates that the playbook was executed successfully on the host 172.17.0.3. The "PLAY RECAP" section shows that 4 tasks were successful, 1 task was changed, and no tasks were unreachable, failed, skipped, rescued, or ignored.