

QUESTION 1

1. Ansible

[< Previous](#)[Next >](#)

You ran an Ansible playbook to install Python3 on multiple servers, but it failed on one of the servers. The error message was:

```
fatal: [server2]: failed! => {"msg": "no package matching 'python3' found available"}
```

Questions:

- i. Why did this error occur?
- ii. How can you modify your playbook to handle this issue dynamically using Ansible Facts?
- iii. Write a playbook snippet that ensures Python3 gets installed correctly on any Linux distribution.

SOLUTION:

i. Why did this error occur?

The error:

```
fatal: [server2]: failed! => {"msg": "no package matching 'python3' found available"}
```

means that the python3 package is not available in the server's package manager repositories under the name python3.

Reasons:

The package name varies across Linux distributions (e.g., python3, python, etc.).

The package manager cache is outdated.

The server uses a minimal installation without enabled repositories.

ii. How can you modify your playbook to handle this issue dynamically using Ansible Facts?

We can use Ansible Facts to detect the OS and conditionally install the correct Python package depending on the distribution (e.g., Debian/Ubuntu, RHEL/CentOS, Amazon Linux, etc.).

iii. Playbook snippet to install Python3 on any Linux distribution

- name: Ensure Python3 is installed on all Linux distros

hosts: all

become: yes

tasks:

- name: Update apt cache

apt:

update_cache: yes

when: ansible_os_family == "Debian"

- name: Install Python3 on Debian-based systems

apt:

name: python3

state: present

when: ansible_os_family == "Debian"

- name: Install Python3 on RedHat

yum:

name: python3

state: present

when: ansible_os_family == "RedHat"

- name: Install Python3 on amazon

amazon.aws.aws_package:

name: python3

state: present

when: ansible_distribution == "Amazon"

- name: Ensure Python3 is available

command: python3 --version

register: python_check

changed_when: false