**Day2-ansible test**

**Q1.**

**i. Why did this error occur?**

"Right, so that fatal error about not finding the 'python3' package on server2? Yeah, that happens because the name 'python3' isn't a universal thing across all Linux systems. Different ones, like Ubuntu and Red Hat, sometimes call it different stuff, or you might need to enable extra software sources on some of them before you can even see it."

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

"The smart way to deal with that is to use what Ansible calls 'Facts'. Basically, Ansible can figure out what kind of operating system it's talking to on each server. So, instead of just blindly trying to install 'python3', you can tell Ansible to check if it's a Debian-based system (like Ubuntu) or a Red Hat one (like CentOS). Then, depending on what it finds, it can use the right command and the right package name for that specific system. This way, you're not just guessing; you're actually adapting to each server."

**iii. Write a playbook snippet that ensures Python3 gets installed correctly on any Linux distribution.**

- name: Ensure Python3 is installed on all Linux servers

hosts: all

become: yes

tasks:

- name: Install Python3 on Debian/Ubuntu

apt:

name: python3

state: present

update\_cache: yes

when: ansible\_os\_family == "Debian"

- name: Install Python3 on RedHat/CentOS/Amazon

yum:

name: python3

state: present

when: ansible\_os\_family == "RedHat"

- name: Install EPEL and Python3 on older RedHat/CentOS if needed

block:

- name: Install EPEL repository

yum:

name: epel-release

state: present

- name: Install Python3 from EPEL

yum:

name: python3

state: present

when: ansible\_distribution in ["CentOS", "RedHat"] and ansible\_distribution\_major\_version | int <= 7

- name: Install Python3 on Alpine

apk:

name: python3

state: present

when: ansible\_os\_family == "Alpine"

screenshots:







