**GENERAL STRUCTURE OF PLAYBOOK**

- name: <Playbook Name>

hosts: <Target Hosts>

become: <yes/no> # (Optional) Run tasks as a privileged user (e.g., root)

vars:

<variable\_name>: <value> # (Optional) Define variables

tasks:

- name: <Task Description>

<module\_name>: # (e.g., apt, yum, file, copy, template, service, etc.)

<module\_parameters>

when: <condition> # (Optional) Run the task only if the condition is met

register: <variable\_name> # (Optional) Store command output

handlers: # (Optional) Define actions to be triggered when notified

- name: <Handler Name>

<module\_name>: # (e.g., service)

<module\_parameters>

1. Install and Start a Web Server (Apache or Nginx)

Answer (`install\_webserver.yml`)

- name: Install and configure web server

hosts: all

become: yes

tasks:

- name: Install Apache on CentOS

yum: *## Used to manage packages on* ***RedHat-based*** *systems.*

name: httpd

state: present

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

- name: Install Nginx on Ubuntu

apt:

name: nginx

state: present

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

- name: Start and enable the web server

service: *## used to manage system services*

name: "{{ 'httpd' if ansible\_os\_family == 'RedHat' else 'nginx' }}"

state: started

enabled: yes

Steps to Execute

1. Save the playbook as `install\_webserver.yml`.

2. Run the playbook:

ansible-playbook -i inventory install\_webserver.yml

Explanation

- Uses `when` conditions to install the correct web server based on OS.

- Dynamically sets the service name (`httpd` for CentOS, `nginx` for Ubuntu).

2. Create a User Only If It Doesn’t Exist

Answer (`create\_user.yml`)

name: Ensure deployuser exists

hosts: all

become: yes

tasks:

- name: Check if user exists

command: id deployuser

register: user\_check

ignore\_errors: yes

name: Create deployuser if it does not exist

user:

name: deployuser

groups: sudo

shell: /bin/bash

state: present

when: user\_check.rc != 0

Steps to Execute

1. Save the file as `create\_user.yml`.

2. Run:

ansible-playbook -i inventory create\_user.yml

Explanation

- Runs `id deployuser` to check if the user exists.

- Creates the user only if it doesn’t exist (`when: user\_check.rc != 0`).

1. Ensure a Service is Running and Restart It on Failure

Answer (`ensure\_nginx.yml`)

- name: Ensure Nginx is running

hosts: all

become: yes

tasks:

- name: Check if Nginx is active

command: systemctl is-active nginx

register: nginx\_status

ignore\_errors: yes

- name: Restart Nginx if not running

service:

name: nginx

state: restarted

when: nginx\_status.rc != 0

Steps to Execute

1. Save as `ensure\_nginx.yml`.

2. Run:

bash

ansible-playbook -i inventory ensure\_nginx.yml

Explanation

- Runs `systemctl is-active nginx` to check if Nginx is active.

- If not active (`rc != 0`), it restarts Nginx.

4. Install Multiple Packages Based on OS

Answer (`install\_packages.yml`)

- name: Install packages based on OS

hosts: all

become: yes

tasks:

- name: Install on Ubuntu

apt:

name:

- vim

- git

state: present

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

- name: Install on CentOS

yum:

name:

- nano

- git

state: present

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

Steps to Execute

1. Save as `install\_packages.yml`.

2. Run:

bash

ansible-playbook -i inventory install\_packages.yml

Explanation

- Uses `when` conditions to install different packages based on OS.