**TASK 1**

**Question:**  
I have a file located at /home/user/data.csv, but we need to copy this file using Ansible without using any Ansible playbook.

**Answer:**  
Here we must use an **Ansible ad-hoc command** with the copy module.

**Command:**

ansible ubuntu -m copy -a "src=/home/user/data.csv dest=/tmp/data.csv" -b

* ubuntu → is the group/host defined in your inventory (your AWS Ubuntu worker).
* src → file on **Ansible master (CentOS VM)**.
* dest → location on the **worker node (Ubuntu EC2)**.
* -b → run with sudo.

**TASK 2**

**Question:**  
We have a requirement to obtain memory details for a remote host using an ad-hoc command.

**Answer:**  
We can use the **setup module** to gather facts and filter memory details.

**Command:**

ansible ubuntu -m setup -a 'filter=ansible\_memtotal\_mb'

This will show the **total memory** in MB.  
If you want all memory-related facts:

ansible ubuntu -m setup -a 'filter=ansible\_memory\*'

**TASK 3**

**Question:**  
As per Task 2, we need to obtain memory details using the "gather facts" feature of Ansible.

**Answer:**  
Here, we’ll write a small **playbook** that gathers memory facts and prints them.

**File: memory\_facts.yml**

---

- name: Gather memory facts from Ubuntu worker

hosts: ubuntu

gather\_facts: yes

tasks:

- name: Print total memory

debug:

msg: "Total Memory: {{ ansible\_memtotal\_mb }} MB"

Run:

ansible-playbook memory\_facts.yml

**TASK 4**

**Question:**  
Copy the application.tar.gz file to a remote endpoint, and once the file is copied, extract it automatically using an Ansible playbook.

**Answer:**  
We can use copy + unarchive.

**File: copy\_extract.yml**

---

- name: Copy and extract application file

hosts: ubuntu

become: yes

tasks:

- name: Copy tar file

copy:

src: /home/user/application.tar.gz

dest: /tmp/application.tar.gz

- name: Extract tar.gz file

unarchive:

src: /tmp/application.tar.gz

dest: /opt/app

remote\_src: yes

Run:

ansible-playbook copy\_extract.yml

**TASK 5**

**Question:**  
We have a requirement to install the httpd package and allow port 80/tcp in the firewall using an Ansible playbook.

**Answer:**  
For Ubuntu → install **apache2**. For firewall we use ufw.

**File: httpd\_firewall.yml**

---

- name: Install Apache and allow port 80

hosts: ubuntu

become: yes

tasks:

- name: Install Apache2

apt:

name: apache2

state: present

update\_cache: yes

- name: Allow port 80 in firewall

ufw:

rule: allow

port: 80

proto: tcp

**TASK 6**

**Question:**  
Install and start the httpd service, copy index.html to the Apache server, and verify if webpage is accessible.

**Answer:**

**File: deploy\_web.yml**

---

- name: Deploy web page with Apache

hosts: ubuntu

become: yes

tasks:

- name: Install Apache2

apt:

name: apache2

state: present

update\_cache: yes

- name: Start and enable Apache2 service

service:

name: apache2

state: started

enabled: yes

- name: Copy index.html

copy:

src: /home/user/index.html

dest: /var/www/html/index.html

- name: Verify webpage

uri:

url: http://{{ inventory\_hostname }}/index.html

return\_content: yes

register: webpage

- name: Show webpage content

debug:

msg: "{{ webpage.content }}"

**TASK 7**

**Question:**  
Create a new user and add an entry in "AllowUsers" section of /etc/ssh/sshd\_config using Ansible.

**Answer:**

**File: ssh\_user.yml**

---

- name: Add user and update sshd\_config

hosts: ubuntu

become: yes

tasks:

- name: Create user

user:

name: newuser

password: "{{ 'Password@123' | password\_hash('sha512') }}"

- name: Add user to AllowUsers in sshd\_config

lineinfile:

path: /etc/ssh/sshd\_config

regexp: '^AllowUsers'

line: "AllowUsers newuser"

create: yes

backup: yes

- name: Restart SSH service

service:

name: ssh

state: restarted

**TASK 8**

**Question:**  
Download a file to a remote endpoint.

**Answer:**

**File: download\_file.yml**

---

- name: Download a file on remote host

hosts: ubuntu

become: yes

tasks:

- name: Download file

get\_url:

url: https://example.com/file.zip

dest: /tmp/file.zip

**TASK 9**

**Question:**  
To send a file to a Windows machine (192.168.0.1) in an Ansible playbook.

**Answer:**  
When working with Windows hosts, Ansible uses **WinRM** for communication and Windows-specific modules (instead of copy, we use win\_copy).

**File: send\_file\_windows.yml**

---

- name: Send file to Windows machine

hosts: windows

vars:

ansible\_user: Administrator

ansible\_password: "MySecurePassword123"

ansible\_connection: winrm

ansible\_winrm\_transport: basic

tasks:

- name: Copy file to Windows

win\_copy:

src: /home/user/data.txt

dest: C:\Users\Administrator\Desktop\data.txt

**Explanation:**

* hosts: windows → in your **inventory**, you should have:

[windows]

192.168.0.1

* win\_copy → Windows version of the copy module.
* src → path on your **Ansible master (CentOS)**.
* dest → path on the **Windows machine**.