# Copy Module in Ansible

## In ansible what is the difference between copy and synchronize

**ChatGPT**

In Ansible, both the **copy** and **synchronize** modules are used for file transfers, but they serve slightly different purposes and have distinct use cases.

**1.copy** Module:

- The **copy** module is used to copy files from the Ansible control machine (the machine where you are running Ansible) to remote target machines.
- It is primarily used for one-time file transfers and does not perform any synchronization or comparison between the source and destination files.
- If you need to transfer files or templates from your Ansible control machine to remote servers without any complex synchronization logic, the **copy** module is a straightforward choice.

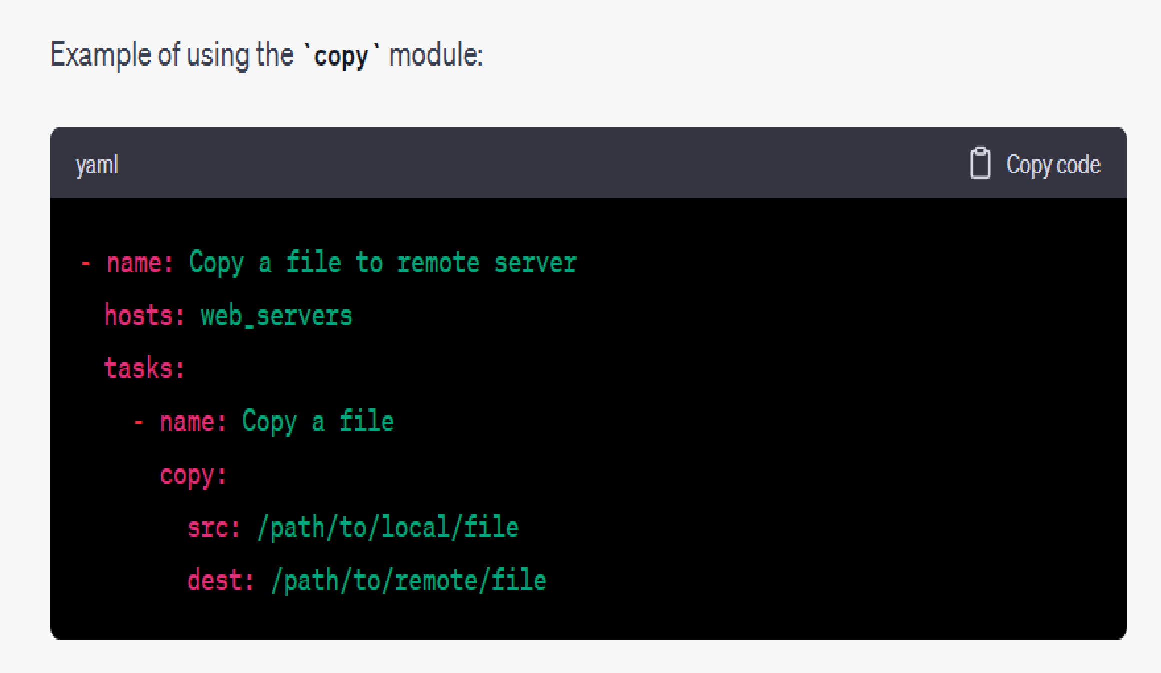
# The synchronize Module in Ansible

**ChatGPT**

**synchronize** Module:

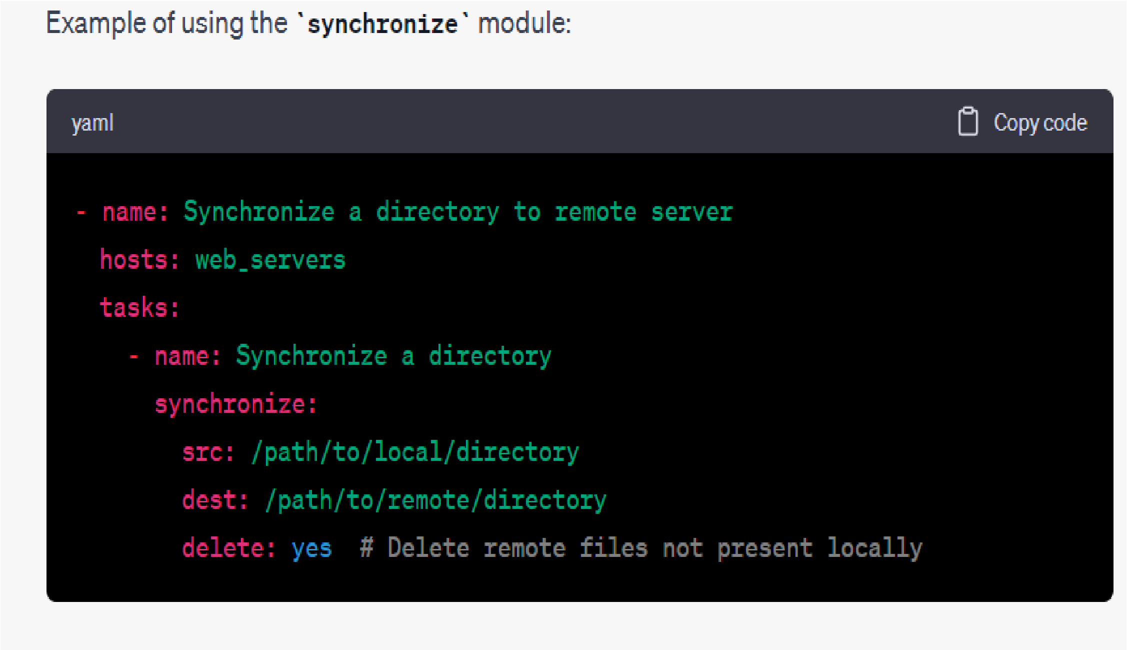
- The **synchronize** module is used for more advanced file synchronization operations between the Ansible control machine and remote target machines.
- It can handle tasks such as mirroring directories, handling deletions, and ensuring that the source and destination directories are in sync.
- It uses the rsync algorithm under the hood, which is efficient for synchronizing files and directories.
- The **synchronize** module is useful when you need to maintain the same directory structure on the remote servers and want to keep them in sync with the local machine, including the ability to delete files on the remote server that are not present locally.

# The Difference between copy and synchronize in Ansible Module



Example of using the `copy` module:

```yaml
- name: Copy a file to remote server
  hosts: web_servers
  tasks:
    - name: Copy a file
      copy:
        src: /path/to/local/file
        dest: /path/to/remote/file
```



Example of using the `synchronize` module:

```yaml
- name: Synchronize a directory to remote server
  hosts: web_servers
  tasks:
    - name: Synchronize a directory
      synchronize:
        src: /path/to/local/directory
        dest: /path/to/remote/directory
        delete: yes  # Delete remote files not present locally
```