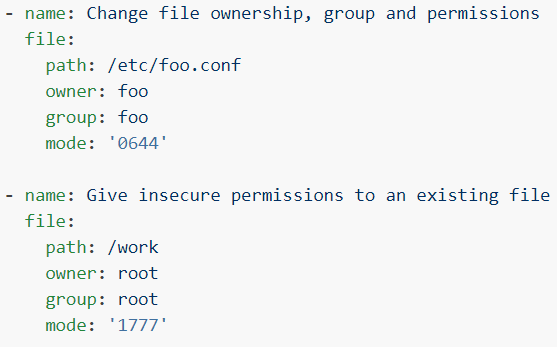
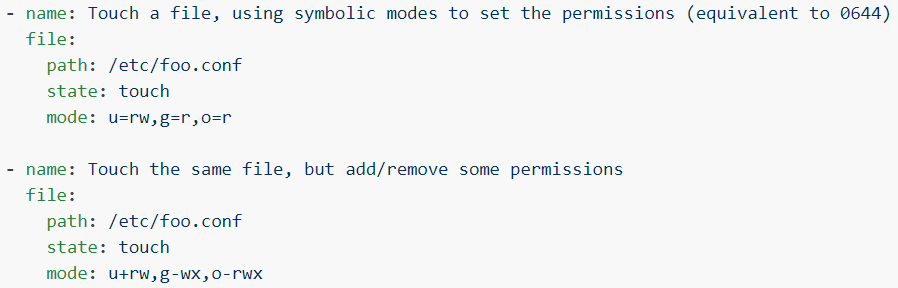
**File module:**

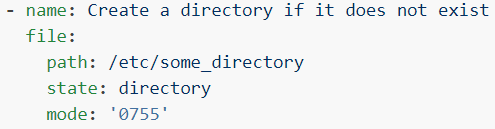
* It used to create, delete files, symlinks or directories.



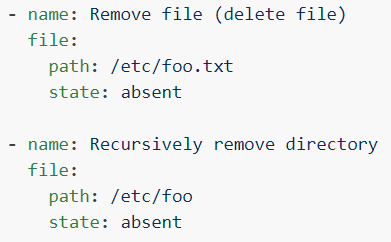
* Changing the file permission and ownership.



* Touch file and adding permissions to in symbolic mode



* Create directory if it doesn’t exist and give permissions.



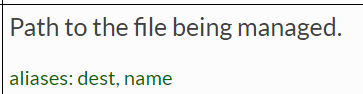
* Remove file or directory.

**Options:**

**owner**

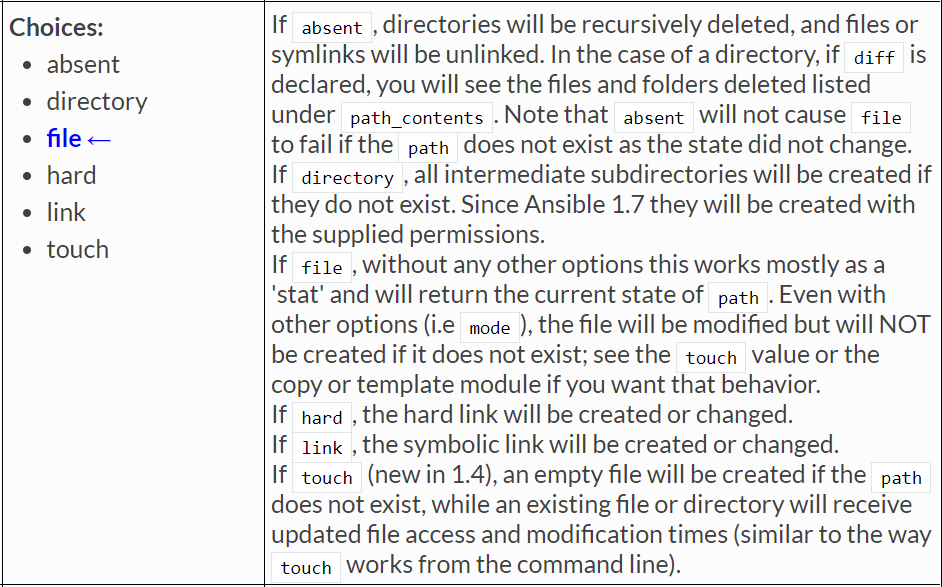


**path**



* Path of the file or directory that we want to create or modify something.

**state**



* If we want to create a file. Use **“touch”.** To create directory. Use **“directory”**
* We should use **“absent”** to delete file or directory by giving the path

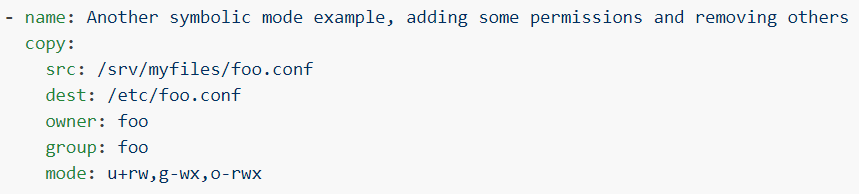
**mode**

* mode is used to set the permission to a file or directory specified in **“path”**. We can either set the permissions in number format or symbolic mode

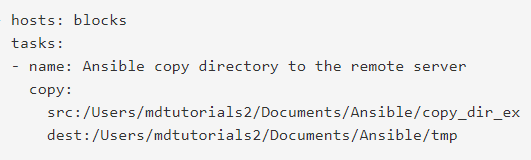
**Copy module:**

* The copy module copies a file or directory from the local or remote machine to a location on the remote machine.
* Use the fetch module to copy files from remote locations to the local box.
* We can use copy module for deployment to copy the war files to tomcat server
* For Windows targets, use the **win\_copy** module instead.

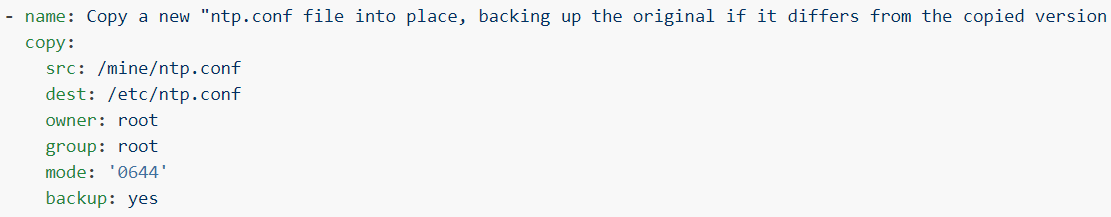




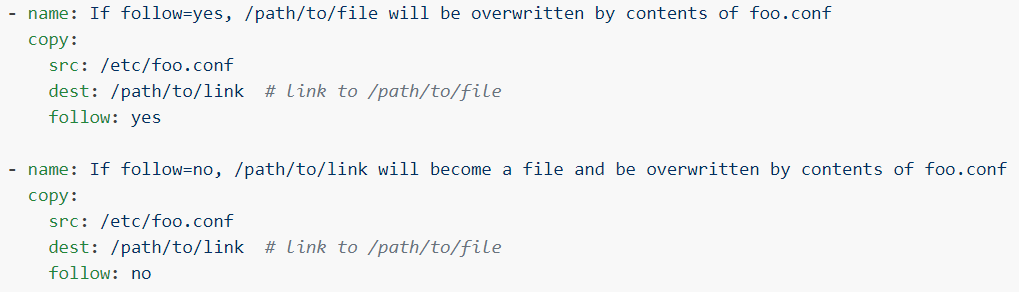
* Copying the files to remote servers with owner, group and permissions.



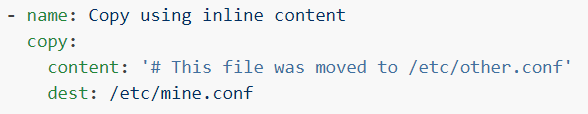
* Copying the directory to remote servers



* Copying the file to remote server by taking the backup if file already exists.



* If follow is yes, then the file inside the given “dest” path will be overwritten by “src” file
* If follow is no, then the path itself will become a file and the content will be overwritten by “src” file.

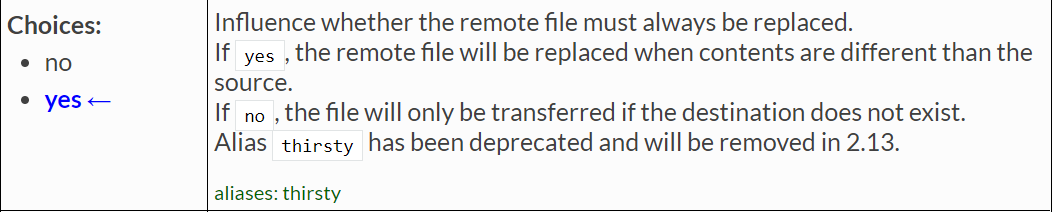


* Adding some content to the remote file

**Options:**

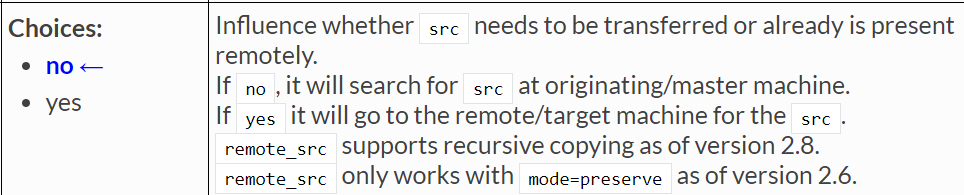
* “src” is the path of local file name and “dest” is the remote server path.
* “mode” is just like the same as we have in “file” module. It is to give the permissions to file or directory with number or symbolic mode

**force**



* If the file is already present in the node servers and the content is same. Then ansible won’t copy the file again.
* If the content is different in node servers than the source file. Then ansible will overwrite that data by default.
* If you want it change this, we can set “force=no”. otherwise, we can ignore it as the value is yes by default.

**remote\_src**



* By default, ansible search for the “src” file in local machine or master server to copy it to the remote servers.
* If we want to copy any file inside the remote server. Then we need to enable “remote\_src” argument

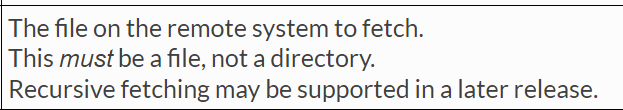
**fetch module:**

* This module works like copy, but in reverse.
* It is used for fetching files from remote machines and storing them locally in a file tree, organized by hostname.
* Files that already exist at dest will be overwritten if they are different than the src.
* This module is also supported for Windows targets



**Options:**

**src**



**dest**

