ansible.builtin.copy module – Copy files to remote locations

Note

This module is part of `ansible-core` and included in all Ansible installations. In most cases, you can use the short module name `copy` even without specifying the `collections:` keyword. However, we recommend you use the FQCN for easy linking to the module documentation and to avoid conflicting with other collections that may have the same module name.

Synopsis

- The `copy` module copies a file from the local or remote machine to a location on the remote machine.

- Use the ansible.builtin.fetch (fetch_module.html#ansible-collections-ansible-builtin-fetch-module) module to copy files from remote locations to the local box.
- If you need variable interpolation in copied files, use the ansible.builtin.template (template_module.html#ansible-collections-ansible-builtin-template-module) module. Using a variable in the `content` field will result in unpredictable output.
- For Windows targets, use the ansible.windows.win_copy (../windows/win_copy_module.html#ansible-collections-ansible-windows-win-copy-module) module instead.

Note

This module has a corresponding action plugin (../../../plugins/action.html#action-plugins).

Parameters

attributes
aliases: attr
string

The attributes the resulting filesystem object should have.

To get supported flags look at the man page for *chattr* on the target system.

This string should contain the attributes in the same order as the one displayed by *lsattr*.

The `=` operator is assumed as default, otherwise `+` or `-` operators need to be included in the string.

backup
boolean

Create a backup file including the timestamp information so you can get the original file back if you somehow clobbered it incorrectly.

Choices:

- **`false`** ← (default)
- `true`

checksum
string

SHA1 checksum of the file being transferred.

Used to validate that the copy of the file was successful.

If this is not provided, ansible will use the local calculated checksum of the src file.

content
string

When used instead of `src`, sets the contents of a file directly to the specified value.

Works only when `dest` is a file. Creates the file if it does not exist.

For advanced formatting or if `content` contains a variable, use the ansible.builtin.template (template_module.html#ansible-collections-ansible-builtin-template-module) module.

decrypt
boolean

This option controls the autodecryption of source files using vault.

Choices:

- `false`
- **`true`** ← (default)

dest
path / required

Remote absolute path where the file should be copied to.

If `src` is a directory, this must be a directory too.

If `dest` is a non-existent path and if either `dest` ends with “/” or `src` is a directory, `dest` is created.

If *dest* is a relative path, the starting directory is determined by the remote host.

If `src` and `dest` are files, the parent directory of `dest` is not created and the task fails if it does not already exist.

directory_mode
any

When doing a recursive copy set the mode for the directories.

If this is not set we will use the system defaults.

The mode is only set on directories which are newly created, and will not affect those that already existed.

follow
boolean

This flag indicates that filesystem links in the destination, if they exist, should be followed.

Choices:

- **`false`** ← (default)
- `true`

force
boolean

Influence whether the remote file must always be replaced.

If `true`, the remote file will be replaced when contents are different than the source.

If `false`, the file will only be transferred if the destination does not exist.

Choices:

- `false`
- **`true`** ← (default)

group
string

Name of the group that should own the filesystem object, as would be fed to *chown*.

When left unspecified, it uses the current group of the current user unless you are root, in which case it can preserve the previous ownership.

local_follow
boolean

This flag indicates that filesystem links in the source tree, if they exist, should be followed.

Choices:

- `false`
- **`true`** ← (default)

mode
any

The permissions of the destination file or directory.

For those used to `/usr/bin/chmod` remember that modes are actually octal numbers. You must either add a leading zero so that Ansible's YAML parser knows it is an octal number (like `0644` or `01777`) or quote it (like `'644'` or `'1777'`) so Ansible receives a string and can do its own conversion from string into number. Giving Ansible a number without following one of these rules will end up with a decimal number which will have unexpected results.

As of Ansible 1.8, the mode may be specified as a symbolic mode (for example, `u+rwx` or `u=rw,g=r,o=r`).

As of Ansible 2.3, the mode may also be the special string `preserve`.

`preserve` means that the file will be given the same permissions as the source file.

When doing a recursive copy, see also `directory_mode`.

If `mode` is not specified and the destination file **does not** exist, the default `umask` on the system will be used when setting the mode for the newly created file.

If `mode` is not specified and the destination file **does** exist, the mode of the existing file will be used.

Specifying `mode` is the best way to ensure files are created with the correct permissions. See CVE-2020-1736 for further details.

owner
string

Name of the user that should own the filesystem object, as would be fed to *chown*.

When left unspecified, it uses the current user unless you are root, in which case it can preserve the previous ownership.

Specifying a numeric username will be assumed to be a user ID and not a username. Avoid numeric usernames to avoid this confusion.

remote_src
boolean

Influence whether `src` needs to be transferred or already is present remotely.

If `false`, it will search for `src` on the controller node.

If `true` it will search for `src` on the managed (remote) node.

`remote_src` supports recursive copying as of version 2.8.

`remote_src` only works with `mode=preserve` as of version 2.6.

Autodecryption of files does not work when `remote_src=yes`.

Choices:

- **`false`** ← (default)
- `true`

selevel
string

The level part of the SELinux filesystem object context.

This is the MLS/MCS attribute, sometimes known as the `range`.

When set to `_default`, it will use the `level` portion of the policy if available.

serole
string

The role part of the SELinux filesystem object context.

When set to `_default`, it will use the `role` portion of the policy if available.

setype
string

The type part of the SELinux filesystem object context.

When set to `_default`, it will use the `type` portion of the policy if available.

seuser
string

The user part of the SELinux filesystem object context.

By default it uses the `system` policy, where applicable.

When set to `_default`, it will use the `user` portion of the policy if available.

src
path

Local path to a file to copy to the remote server.

This can be absolute or relative.

If path is a directory, it is copied recursively. In this case, if path ends with “/”, only inside contents of that directory are copied to destination. Otherwise, if it does not end with “/”, the directory itself with all contents is copied. This behavior is similar to the `rsync` command line tool.

unsafe_writes
boolean

Influence when to use atomic operation to prevent data corruption or inconsistent reads from the target filesystem object.

By default this module uses atomic operations to prevent data corruption or inconsistent reads from the target filesystem objects, but sometimes systems are configured or just broken in ways that prevent this. One example is docker mounted filesystem objects, which cannot be updated atomically from inside the container and can only be written in an unsafe manner.

This option allows Ansible to fall back to unsafe methods of updating filesystem objects when atomic operations fail (however, it doesn't force Ansible to perform unsafe writes).

IMPORTANT! Unsafe writes are subject to race conditions and can lead to data corruption.

Choices:

- `false` ← (default)
- `true`

validate
string

The validation command to run before copying the updated file into the final destination.

A temporary file path is used to validate, passed in through '%s' which must be present as in the examples below.

Also, the command is passed securely so shell features such as expansion and pipes will not work.

For an example on how to handle more complex validation than what this option provides, see handling complex validation (../../../reference_appendices/faq.html#complex-configuration-validation).

Attributes

action

Support: full

Indicates this has a corresponding action plugin so some parts of the options can be executed on the controller

async

Support: none

Supports being used with the `async` keyword

bypass_host_loop

Support: none

Forces a 'global' task that does not execute per host, this bypasses per host templating and serial, throttle and other loop considerations

Conditionals will work as if `run_once` is being used, variables used will be from the first available host

This action will not work normally outside of lockstep strategies

check_mode

Support: full

Can run in check_mode and return changed status prediction without modifying target

diff_mode

Support: full

Will return details on what has changed (or possibly needs changing in check_mode), when in diff mode

platform

Platform: posix

Target OS/families that can be operated against

safe_file_operations

Support: full

Uses Ansible's strict file operation functions to ensure proper permissions and avoid data corruption

vault

Support: full

Can automatically decrypt Ansible vaulted files

Notes

Note

- The ansible.builtin.copy module recursively copy facility does not scale to lots (>hundreds) of files.

See Also

See also

ansible.builtin.assemble (assemble_module.html#ansible-collections-ansible-builtin-assemble-module)
Assemble configuration files from fragments.

ansible.builtin.fetch (fetch_module.html#ansible-collections-ansible-builtin-fetch-module)
Fetch files from remote nodes.

ansible.builtin.file (file_module.html#ansible-collections-ansible-builtin-file-module)
Manage files and file properties.

ansible.builtin.template (template_module.html#ansible-collections-ansible-builtin-template-module)

Template a file out to a target host.

ansible.posix.synchronize (../posix/synchronize_module.html#ansible-collections-ansible-posix-synchronize-module)

A wrapper around rsync to make common tasks in your playbooks quick and easy.

ansible.windows.win_copy (../windows/win_copy_module.html#ansible-collections-ansible-windows-win-copy-module)

Copies files to remote locations on windows hosts.

Examples

```
- name: Copy file with owner and permissions
  ansible.builtin.copy:
    src: /srv/myfiles/foo.conf
    dest: /etc/foo.conf
    owner: foo
    group: foo
    mode: '0644'

- name: Copy file with owner and permission, using symbolic representation
  ansible.builtin.copy:
    src: /srv/myfiles/foo.conf
    dest: /etc/foo.conf
    owner: foo
    group: foo
    mode: u=rw,g=r,o=r

- name: Another symbolic mode example, adding some permissions and removing others
  ansible.builtin.copy:
    src: /srv/myfiles/foo.conf
    dest: /etc/foo.conf
    owner: foo
    group: foo
    mode: u+rw,g-wx,o-rwx

- name: Copy a new "ntp.conf" file into place, backing up the original if it differs
from the copied version
  ansible.builtin.copy:
    src: /mine/ntp.conf
    dest: /etc/ntp.conf
    owner: root
    group: root
    mode: '0644'
    backup: yes

- name: Copy a new "sudoers" file into place, after passing validation with visudo
  ansible.builtin.copy:
    src: /mine/sudoers
    dest: /etc/sudoers
    validate: /usr/sbin/visudo -csf %s

- name: Copy a "sudoers" file on the remote machine for editing
  ansible.builtin.copy:
    src: /etc/sudoers
    dest: /etc/sudoers.edit
    remote_src: yes
    validate: /usr/sbin/visudo -csf %s

- name: Copy using inline content
  ansible.builtin.copy:
    content: '# This file was moved to /etc/other.conf'
    dest: /etc/mine.conf

- name: If follow=yes, /path/to/file will be overwritten by contents of foo.conf
  ansible.builtin.copy:
    src: /etc/foo.conf
    dest: /path/to/link  # link to /path/to/file
    follow: yes

- name: If follow=no, /path/to/link will become a file and be overwritten by contents
of foo.conf
```

```
ansible.builtin.copy:
  src: /etc/foo.conf
  dest: /path/to/link  # link to /path/to/file
  follow: no
```

Return Values

Common return values are documented here (../../../reference_appendices/common_return_values.html#common-return-values), the following are the fields unique to this module:

backup_file
string

Name of backup file created.

Returned: changed and if backup=yes

Sample: `"/path/to/file.txt.2015-02-12@22:09~"`

checksum
string

SHA1 checksum of the file after running copy.

Returned: success

Sample: `"6e642bb8dd5c2e027bf21dd923337cbb4214f827"`

dest
string

Destination file/path.

Returned: success

Sample: `"/path/to/file.txt"`

gid
integer

Group id of the file, after execution.

Returned: success

Sample: `100`

group
string

Group of the file, after execution.

Returned: success

Sample: `"httpd"`

md5sum
string

MD5 checksum of the file after running copy.

Returned: when supported

Sample: `"2a5aeecc61dc98c4d780b14b330e3282"`

mode
string

Permissions of the target, after execution.

Returned: success

Sample: `"0644"`

owner
string

Owner of the file, after execution.

Returned: success

Sample: `"httpd"`

size
integer

Size of the target, after execution.

Returned: success

Sample: `1220`

src
string

Source file used for the copy on the target machine.

Returned: changed

Sample: `"/home/httpd/.ansible/tmp/ansible-tmp-1423796390.97-147729857856000/source"`

state
string

State of the target, after execution.

Returned: success

Sample: `"file"`

uid
integer

Owner id of the file, after execution.

Returned: success

Sample: `100`

Authors

- Ansible Core Team

- Michael DeHaan

Collection links

Issue Tracker (https://github.com/ansible/ansible/issues)

Repository (Sources) (https://github.com/ansible/ansible)

Communication (./#communication-for-ansible-builtin)