ansible.builtin.template module – Template a file out to a target host

Note

This module is part of `ansible-core` and included in all Ansible installations. In most cases, you can use the short module name `template` 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

- Templates are processed by the Jinja2 templating language (http://jinja.pocoo.org/docs/).
- Documentation on the template formatting can be found in the Template Designer Documentation (http://jinja.pocoo.org/docs/templates/).

- Additional variables listed below can be used in templates.
- `ansible_managed` (configurable via the `defaults` section of `ansible.cfg`) contains a string which can be used to describe the template name, host, modification time of the template file and the owner uid.
- `template_host` contains the node name of the template's machine.
- `template_uid` is the numeric user id of the owner.
- `template_path` is the path of the template.
- `template_fullpath` is the absolute path of the template.
- `template_destpath` is the path of the template on the remote system (added in 2.8).
- `template_run_date` is the date that the template was rendered.

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`

block_end_string
string

The string marking the end of a block.

Default: `"%}"`

block_start_string
string

The string marking the beginning of a block.

Default: `"{%"`

comment_end_string
string
added in ansible-core 2.12

The string marking the end of a comment statement.

comment_start_string
string
added in ansible-core 2.12

The string marking the beginning of a comment statement.

dest
path / required

Location to render the template to on the remote machine.

follow
boolean

Determine whether symbolic links should be followed.

When set to `true` symbolic links will be followed, if they exist.

When set to `false` symbolic links will not be followed.

Previous to Ansible 2.4, this was hardcoded as `true`.

Choices:

- **`false`** ← (default)
- `true`

force
boolean

Determine when the file is being transferred if the destination already exists.

When set to `yes`, replace the remote file when contents are different than the source.

When set to `no`, 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.

lstrip_blocks
boolean

Determine when leading spaces and tabs should be stripped.

When set to `yes` leading spaces and tabs are stripped from the start of a line to a block.

Choices:

- **`false`** ← (default)
- `true`

mode
any

The permissions the resulting filesystem object should have.

For those used to */usr/bin/chmod* remember that modes are actually octal numbers. You must give Ansible enough information to parse them correctly. For consistent results, quote octal numbers (for example, `'644'` or `'1777'`) so Ansible receives a string and can do its own conversion from string into number. Adding a leading zero (for example, `0755`) works sometimes, but can fail in loops and some other circumstances.

Giving Ansible a number without following either 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`).

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

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

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

newline_sequence
string

Specify the newline sequence to use for templating files.

Choices:

- **`"\\n"`** ← (default)
- `"\\r"`
- `"\\r\\n"`

output_encoding
string
added in Ansible 2.7

Overrides the encoding used to write the template file defined by `dest`.

It defaults to `utf-8`, but any encoding supported by python can be used.

The source template file must always be encoded using `utf-8`, for homogeneity.

Default: `"utf-8"`

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.

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 / required

Path of a Jinja2 formatted template on the Ansible controller.

This can be a relative or an absolute path.

The file must be encoded with `utf-8` but *output_encoding* can be used to control the encoding of the output template.

trim_blocks
boolean

Determine when newlines should be removed from blocks.

When set to `yes` the first newline after a block is removed (block, not variable tag!).

Choices:

- `false`
- **`true`** ← (default)

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).

variable_end_string
string

The string marking the end of a print statement.

Default: `"}}"`

variable_start_string
string

The string marking the beginning of a print statement.

Default: `"{{"`

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

- For Windows you can use ansible.windows.win_template (../windows/win_template_module.html#ansible-collections-ansible-windows-win-template-module) which uses `\r\n` as `newline_sequence` by default.
- The `jinja2_native` setting has no effect. Native types are never used in the `template` module which is by design used for generating text files. For working with templates and utilizing Jinja2 native types see the `jinja2_native` parameter of the `template`

`lookup`.

- Including a string that uses a date in the template will result in the template being marked 'changed' each time.
- Since Ansible 0.9, templates are loaded with `trim_blocks=True`.
- Also, you can override jinja2 settings by adding a special header to template file. i.e. `#jinja2:variable_start_string:'[%', variable_end_string:'%]', trim_blocks: False` which changes the variable interpolation markers to `[% var %]` instead of `{{ var }}`. This is the best way to prevent evaluation of things that look like, but should not be Jinja2.
- To find Byte Order Marks in files, use `Format-Hex <file> -Count 16` on Windows, and use `od -a -t x1 -N 16 <file>` on Linux.

See Also

See also

ansible.builtin.copy (copy_module.html#ansible-collections-ansible-builtin-copy-module)
Copy files to remote locations.

ansible.windows.win_copy (../windows/win_copy_module.html#ansible-collections-ansible-windows-win-copy-module)
Copies files to remote locations on windows hosts.

ansible.windows.win_template (../windows/win_template_module.html#ansible-collections-ansible-windows-win-template-module)
Template a file out to a remote server.

Examples

```
- name: Template a file to /etc/file.conf
  ansible.builtin.template:
    src: /mytemplates/foo.j2
    dest: /etc/file.conf
    owner: bin
    group: wheel
    mode: '0644'

- name: Template a file, using symbolic modes (equivalent to 0644)
  ansible.builtin.template:
    src: /mytemplates/foo.j2
    dest: /etc/file.conf
    owner: bin
    group: wheel
    mode: u=rw,g=r,o=r

- name: Copy a version of named.conf that is dependent on the OS. setype obtained by
doing ls -Z /etc/named.conf on original file
  ansible.builtin.template:
    src: named.conf_{{ ansible_os_family }}.j2
    dest: /etc/named.conf
    group: named
    setype: named_conf_t
    mode: 0640

- name: Create a DOS-style text file from a template
  ansible.builtin.template:
    src: config.ini.j2
    dest: /share/windows/config.ini
    newline_sequence: '\r\n'

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

- name: Update sshd configuration safely, avoid locking yourself out
  ansible.builtin.template:
    src: etc/ssh/sshd_config.j2
    dest: /etc/ssh/sshd_config
    owner: root
    group: root
    mode: '0600'
    validate: /usr/sbin/sshd -t -f %s
    backup: yes
```

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)