Validating tasks: check mode and diff mode

Ansible provides two modes of execution that validate tasks: check mode and diff mode. These modes can be used separately or together. They are useful when you are creating or editing a playbook or role and you want to know what it will do. In check mode, Ansible runs without making any changes on remote systems. Modules that support check mode report the changes they would have made. Modules that do not support check mode report nothing and do nothing. In diff mode, Ansible provides before-and-after comparisons. Modules that support diff mode display detailed information. You can combine check mode and diff mode for detailed validation of your playbook or role.

Using check mode

Check mode is just a simulation. It will not generate output for tasks that use :ref:`conditionals based on registered variables <conditionals_registered_vars>` (results of prior tasks). However, it is great for validating configuration management playbooks that run on one node at a time. To run a playbook in check mode:

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\user_guide\[ansible-devel][docs][docsite][rst][user_guide]playbooks_checkmode.rst`, line 15); *backlink*

Unknown interpreted text role "ref".

```
ansible-playbook foo.yml --check
```

Enforcing or preventing check mode on tasks

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\user_guide\[ansible-devel][docs][docsite][rst][user_guide]playbooks_checkmode.rst`, line 26)

Unknown directive type "versionadded".

```
.. versionadded:: 2.2
```

If you want certain tasks to run in check mode always, or never, regardless of whether you run the playbook with or without `--check`, you can add the `check_mode` option to those tasks:

- To force a task to run in check mode, even when the playbook is called without `--check`, set `check_mode: yes`.
- To force a task to run in normal mode and make changes to the system, even when the playbook is called with `--check`, set `check_mode: no`.

For example:

```
tasks:
  - name: This task will always make changes to the system
    ansible.builtin.command: /something/to/run --even-in-check-mode
    check_mode: no

  - name: This task will never make changes to the system
    ansible.builtin.lineinfile:
      line: "important config"
      dest: /path/to/myconfig.conf
      state: present
    check_mode: yes
    register: changes_to_important_config
```

Running single tasks with `check_mode: yes` can be useful for testing Ansible modules, either to test the module itself or to test the conditions under which a module would make changes. You can register variables (see :ref:`playbooks_conditionals`) on these tasks for even more detail on the potential changes.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\user_guide\[ansible-devel][docs][docsite][rst]`

[user_guide]playbooks_checkmode.rst, line 50); *backlink*

Unknown interpreted text role "ref".

Note

Prior to version 2.2 only the equivalent of `check_mode: no` existed. The notation for that was `always_run: yes`.

Skipping tasks or ignoring errors in check mode

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\user_guide\[ansible-devel][docs][docsite][rst][user_guide]playbooks_checkmode.rst, line 57)

Unknown directive type "versionadded".

```
.. versionadded:: 2.1
```

If you want to skip a task or ignore errors on a task when you run Ansible in check mode, you can use a boolean magic variable `ansible_check_mode`, which is set to `True` when Ansible runs in check mode. For example:

```
tasks:

  - name: This task will be skipped in check mode
    ansible.builtin.git:
      repo: ssh://git@github.com/mylogin/hello.git
      dest: /home/mylogin/hello
    when: not ansible_check_mode

  - name: This task will ignore errors in check mode
    ansible.builtin.git:
      repo: ssh://git@github.com/mylogin/hello.git
      dest: /home/mylogin/hello
    ignore_errors: "{{ ansible_check_mode }}"
```

Using diff mode

The `--diff` option for ansible-playbook can be used alone or with `--check`. When you run in diff mode, any module that supports diff mode reports the changes made or, if used with `--check`, the changes that would have been made. Diff mode is most common in modules that manipulate files (for example, the template module) but other modules might also show 'before and after' information (for example, the user module).

Diff mode produces a large amount of output, so it is best used when checking a single host at a time. For example:

```
ansible-playbook foo.yml --check --diff --limit foo.example.com
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\user_guide\[ansible-devel][docs][docsite][rst][user_guide]playbooks_checkmode.rst, line 90)

Unknown directive type "versionadded".

```
.. versionadded:: 2.4
```

Enforcing or preventing diff mode on tasks

Because the `--diff` option can reveal sensitive information, you can disable it for a task by specifying `diff: no`. For example:

```
tasks:
  - name: This task will not report a diff when the file changes
    ansible.builtin.template:
      src: secret.conf.j2
      dest: /etc/secret.conf
      owner: root
      group: root
      mode: '0600'
    diff: no
```