Ansible 2.7 Porting Guide

This section discusses the behavioral changes between Ansible 2.6 and Ansible 2.7.

It is intended to assist in updating your playbooks, plugins and other parts of your Ansible infrastructure so they will work with this version of Ansible.

We suggest you read this page along with Ansible Changelog for 2.7 to understand what updates you may need to make.

This document is part of a collection on porting. The complete list of porting guides can be found at :ref:`porting guides <porting_guides>`.

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\porting_guides\(ansible-devel)(docs)(docsite)(rst)(porting_guides)porting_guide_2.7.rst, line 13); *backlink***
>
> Unknown interpreted text role "ref".

Topics

Command Line

If you specify `--tags` or `--skip-tags` multiple times on the command line, Ansible will merge the specified tags together. In previous versions of Ansible, you could set `merge_multiple_cli_tags` to `False` if you wanted to keep only the last-specified `--tags`. This config option existed for backwards compatibility. The overwriting behavior was deprecated in 2.3 and the default behavior was changed in 2.4. Ansible-2.7 removes the config option; multiple `--tags` are now always merged.

If you have a shell script that depends on setting `merge_multiple_cli_tags` to `False`, please upgrade your script so it only adds the `--tags` you actually want before upgrading to Ansible-2.7.

Python Compatibility

Ansible has dropped compatibility with Python-2.6 on the controller (The host where :command:`/usr/bin/ansible` or :command:`/usr/bin/ansible-playbook` is run). Modules shipped with Ansible can still be used to manage hosts which only have Python-2.6. You just need to have a host with Python-2.7 or Python-3.5 or greater to manage those hosts from.

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\porting_guides\(ansible-devel)(docs)(docsite)(rst)(porting_guides)porting_guide_2.7.rst, line 34); *backlink***
>
> Unknown interpreted text role "command".

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\porting_guides\(ansible-devel)(docs)(docsite)(rst)(porting_guides)porting_guide_2.7.rst, line 34); *backlink***
>
> Unknown interpreted text role "command".

One thing that this does affect is the ability to use :command:`/usr/bin/ansible-pull` to manage a host which has Python-2.6. `ansible-pull` runs on the host being managed but it is a controller script, not a module so it will need an updated Python. Actively developed Linux distros which ship with Python-2.6 have some means to install newer Python versions (For instance, you can install Python-2.7 via an SCL on RHEL-6) but you may need to also install Python bindings for many common modules to work (For RHEL-6, for instance, selinux bindings and yum would have to be installed for the updated Python install).

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\porting_guides\(ansible-devel)(docs)(docsite)(rst)(porting_guides)porting_guide_2.7.rst, line 39); *backlink***
>
> Unknown interpreted text role "command".

The decision to drop Python-2.6 support on the controller was made because many dependent libraries are becoming unavailable there. In particular, python-cryptography is no longer available for Python-2.6 and the last release of pycrypto (the alternative to python-cryptography) has known security bugs which will never be fixed.

Playbook

Role Precedence Fix during Role Loading

Ansible 2.7 makes a small change to variable precedence when loading roles, resolving a bug, ensuring that role loading matches :ref:`variable precedence expectations <ansible_variable_precedence>`.

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\porting_guides\(ansible-devel)(docs)(docsite)(rst)(porting_guides)porting_guide_2.7.rst, line 59); *backlink***
>
> Unknown interpreted text role "ref".

Before Ansible 2.7, when loading a role, the variables defined in the role's `vars/main.yml` and `defaults/main.yml` were not available when parsing the role's `tasks/main.yml` file. This prevented the role from utilizing these variables when being parsed. The problem manifested when `import_tasks` or `import_role` was used with a variable defined in the role's vars or defaults.

In Ansible 2.7, role `vars` and `defaults` are now parsed before `tasks/main.yml`. This can cause a change in behavior if the same variable is defined at the play level and the role level with different values, and used in `import_tasks` or `import_role` to define the role or file to import.

include_role and import_role variable exposure

In Ansible 2.7 a new module argument named `public` was added to the `include_role` module that dictates whether or not the role's `defaults` and `vars` will be exposed outside of the role, allowing those variables to be used by later tasks. This value defaults to `public: False`, matching current behavior.

`import_role` does not support the `public` argument, and will unconditionally expose the role's `defaults` and `vars` to the rest of the playbook. This functionality brings `import_role` into closer alignment with roles listed within the `roles` header in a play.

There is an important difference in the way that `include_role` (dynamic) will expose the role's variables, as opposed to `import_role` (static). `import_role` is a pre-processor, and the `defaults` and `vars` are evaluated at playbook parsing, making the variables available to tasks and roles listed at any point in the play. `include_role` is a conditional task, and the `defaults` and `vars` are evaluated at execution time, making the variables available to tasks and roles listed *after* the `include_role` task.

include_tasks/import_tasks inline variables

As of Ansible 2.7, *include_tasks* and *import_tasks* can no longer accept inline variables. Instead of using inline variables, tasks should supply variables under the `vars` keyword.

OLD In Ansible 2.6 (and earlier) the following was valid syntax for specifying variables:

```
- include_tasks: include_me.yml variable=value
```

NEW In Ansible 2.7 the task should be changed to use the `vars` keyword:

```
- include_tasks: include_me.yml
  vars:
    variable: value
```

vars_prompt with unknown algorithms

vars_prompt now throws an error if the hash algorithm specified in encrypt is not supported by the controller. This increases the safety of vars_prompt as it previously returned None if the algorithm was unknown. Some modules, notably the user module, treated

a password of None as a request not to set a password. If your playbook starts erroring because of this, change the hashing algorithm being used with this filter.

Deprecated

Expedited Deprecation: Use of `__file__` in `AnsibleModule`

> **Note**
>
> The use of the `__file__` variable is deprecated in Ansible 2.7 and **will be eliminated in Ansible 2.8**. This is much quicker than our usual 4-release deprecation cycle.

We are deprecating the use of the `__file__` variable to refer to the file containing the currently-running code. This common Python technique for finding a filesystem path does not always work (even in vanilla Python). Sometimes a Python module can be imported from a virtual location (like inside of a zip file). When this happens, the `__file__` variable will reference a virtual location pointing to inside of the zip file. This can cause problems if, for instance, the code was trying to use `__file__` to find the directory containing the python module to write some temporary information.

Before the introduction of AnsiBallZ in Ansible 2.1, using `__file__` worked in `AnsibleModule` sometimes, but any module that used it would fail when pipelining was turned on (because the module would be piped into the python interpreter's standard input, so `__file__` wouldn't contain a file path). AnsiBallZ unintentionally made using `__file__` work, by always creating a temporary file for `AnsibleModule` to reside in.

Ansible 2.8 will no longer create a temporary file for `AnsibleModule`; instead it will read the file out of a zip file. This change should speed up module execution, but it does mean that starting with Ansible 2.8, referencing `__file__` will always fail in `AnsibleModule`.

If you are the author of a third-party module which uses `__file__` with `AnsibleModule`, please update your module(s) now, while the use of `__file__` is deprecated but still available. The most common use of `__file__` is to find a directory to write a temporary file. In Ansible 2.5 and above, you can use the `tmpdir` attribute on an `AnsibleModule` instance instead, as shown in this code from the :ref:`apt module <ansible_2_7:apt_module>`:

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\porting_guides\(ansible-devel)(docs)(docsite)(rst)(porting_guides)porting_guide_2.7.rst`, line 117); *backlink***
>
> Unknown interpreted text role "ref".

```
-    tempdir = os.path.dirname(__file__)
-    package = os.path.join(tempdir, to_native(deb.rsplit('/', 1)[1]))
+    package = os.path.join(module.tmpdir, to_native(deb.rsplit('/', 1)[1]))
```

Using a loop on a package module via squash_actions

The use of `squash_actions` to invoke a package module, such as "yum", to only invoke the module once is deprecated, and will be removed in Ansible 2.11.

Instead of relying on implicit squashing, tasks should instead supply the list directly to the `name`, `pkg` or `package` parameter of the module. This functionality has been supported in most modules since Ansible 2.3.

OLD In Ansible 2.6 (and earlier) the following task would invoke the "yum" module only 1 time to install multiple packages

```
- name: Install packages
  yum:
    name: "{{ item }}"
    state: present
  with_items: "{{ packages }}"
```

NEW In Ansible 2.7 it should be changed to look like this:

```
- name: Install packages
  yum:
    name: "{{ packages }}"
    state: present
```

Modules

Major changes in popular modules are detailed here

- The :ref:`DEFAULT_SYSLOG_FACILITY` configuration option tells Ansible modules to use a specific syslog facility when logging information on all managed machines. Due to a bug with older Ansible versions, this setting did not affect machines

using journald with the systemd Python bindings installed. On those machines, Ansible log messages were sent to `/var/log/messages`, even if you set :ref:`DEFAULT_SYSLOG_FACILITY`. Ansible 2.7 fixes this bug, routing all Ansible log messages according to the value set for :ref:`DEFAULT_SYSLOG_FACILITY`. If you have :ref:`DEFAULT_SYSLOG_FACILITY` configured, the location of remote logs on systems which use journald may change.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\porting_guides\(ansible-devel)(docs)(docsite)(rst)(porting_guides)porting_guide_2.7.rst, line 158); *backlink*

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\porting_guides\(ansible-devel)(docs)(docsite)(rst)(porting_guides)porting_guide_2.7.rst, line 158); *backlink*

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\porting_guides\(ansible-devel)(docs)(docsite)(rst)(porting_guides)porting_guide_2.7.rst, line 158); *backlink*

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\porting_guides\(ansible-devel)(docs)(docsite)(rst)(porting_guides)porting_guide_2.7.rst, line 158); *backlink*

Unknown interpreted text role "ref".

Modules removed

The following modules no longer exist:

Deprecation notices

The following modules will be removed in Ansible 2.11. Please update your playbooks accordingly.

- `na_cdot_aggregate` use :ref:`na_ontap_aggregate <ansible_2_7:na_ontap_aggregate_module>` instead.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\porting_guides\(ansible-devel)(docs)(docsite)(rst)(porting_guides)porting_guide_2.7.rst, line 178); *backlink*

Unknown interpreted text role "ref".

- `na_cdot_license` use :ref:`na_ontap_license <ansible_2_7:na_ontap_license_module>` instead.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\porting_guides\(ansible-devel)(docs)(docsite)(rst)(porting_guides)porting_guide_2.7.rst, line 179); *backlink*

Unknown interpreted text role "ref".

- `na_cdot_lun` use :ref:`na_ontap_lun <ansible_2_7:na_ontap_lun_module>` instead.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\porting_guides\(ansible-devel)(docs)(docsite)(rst)(porting_guides)porting_guide_2.7.rst, line 180); *backlink*

Unknown interpreted text role "ref".

- `na_cdot_qtree` use :ref:`na_ontap_qtree <ansible_2_7:na_ontap_qtree_module>` instead.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\porting_guides\(ansible-devel)(docs)(docsite)(rst)(porting_guides)porting_guide_2.7.rst, line 181); *backlink*

Unknown interpreted text role "ref".

- `na_cdot_svm` use :ref:`na_ontap_svm <ansible_2_7:na_ontap_svm_module>` instead.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\porting_guides\(ansible-devel)(docs)(docsite)(rst)(porting_guides)porting_guide_2.7.rst`, line 182); *backlink*

Unknown interpreted text role "ref".

- `na_cdot_user` use :ref:`na_ontap_user <ansible_2_7:na_ontap_user_module>` instead.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\porting_guides\(ansible-devel)(docs)(docsite)(rst)(porting_guides)porting_guide_2.7.rst`, line 183); *backlink*

Unknown interpreted text role "ref".

- `na_cdot_user_role` use :ref:`na_ontap_user_role <ansible_2_7:na_ontap_user_role_module>` instead.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\porting_guides\(ansible-devel)(docs)(docsite)(rst)(porting_guides)porting_guide_2.7.rst`, line 184); *backlink*

Unknown interpreted text role "ref".

- `na_cdot_volume` use :ref:`na_ontap_volume <ansible_2_7:na_ontap_volume_module>` instead.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\porting_guides\(ansible-devel)(docs)(docsite)(rst)(porting_guides)porting_guide_2.7.rst`, line 185); *backlink*

Unknown interpreted text role "ref".

- `sf_account_manager` use :ref:`na_elementsw_account<ansible_2_7:na_elementsw_account_module>` instead.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\porting_guides\(ansible-devel)(docs)(docsite)(rst)(porting_guides)porting_guide_2.7.rst`, line 186); *backlink*

Unknown interpreted text role "ref".

- `sf_check_connections` use :ref:`na_elementsw_check_connections<ansible_2_7:na_elementsw_check_connections_module>` instead.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\porting_guides\(ansible-devel)(docs)(docsite)(rst)(porting_guides)porting_guide_2.7.rst`, line 187); *backlink*

Unknown interpreted text role "ref".

- `sf_snapshot_schedule_manager` use :ref:`na_elementsw_snapshot_schedule<ansible_2_7:na_elementsw_snapshot_schedule_module>` instead.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\porting_guides\(ansible-devel)(docs)(docsite)(rst)(porting_guides)porting_guide_2.7.rst`, line 188); *backlink*

Unknown interpreted text role "ref".

- `sf_volume_access_group_manager` use :ref:`na_elementsw_access_group<ansible_2_7:na_elementsw_access_group_module>` instead.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\porting_guides\(ansible-devel)(docs)`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\porting_guides\(ansible-devel)(docs)(docsite)(rst)(porting_guides)porting_guide_2.7.rst, line 189); *backlink*

Unknown interpreted text role "ref".

- `sf_volume_manager` use :ref:`na_elementsw_volume<ansible_2_7:na_elementsw_volume_module>` instead.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\porting_guides\(ansible-devel)(docs)(docsite)(rst)(porting_guides)porting_guide_2.7.rst, line 190); *backlink*

Unknown interpreted text role "ref".

Noteworthy module changes

- Check mode is now supported in the `command` and `shell` modules. However, only when `creates` or `removes` is specified. If either of these are specified, the module will check for existence of the file and report the correct changed status, if they are not included the module will skip like it had done previously.
- The `win_chocolatey` module originally required the `proxy_username` and `proxy_password` to escape any double quotes in the value. This is no longer required and the escaping may cause further issues.
- The `win_uri` module has removed the deprecated option `use_basic_parsing`, since Ansible 2.5 this option did nothing
- The `win_scheduled_task` module has removed the following deprecated options:
  - `executable`, use `path` in an actions entry instead
  - `argument`, use `arguments` in an actions entry instead
  - `store_password`, set `logon_type: password` instead
  - `days_of_week`, use `monthlydow` in a triggers entry instead
  - `frequency`, use `type`, in a triggers entry instead
  - `time`, use `start_boundary` in a triggers entry instead
- The `interface_name` module option for `na_ontap_net_vlan` has been removed and should be removed from your playbooks
- The `win_disk_image` module has deprecated the return value `mount_path`, use `mount_paths[0]` instead. This will be removed in Ansible 2.11.
- `include_role` and `include_tasks` can now be used directly from `ansible` (adhoc) and `ansible-console`:

```
#> ansible -m include_role -a 'name=myrole' all
```

- The `pip` module has added a dependency on `setuptools` to support version requirements, this requirement is for the Python interpreter that executes the module and not the Python interpreter that the module is managing.
- Prior to Ansible 2.7.10, the `replace` module did the opposite of what was intended when using the `before` and `after` options together. This now works properly but may require changes to tasks.

Plugins

- The hash_password filter now throws an error if the hash algorithm specified is not supported by the controller. This increases the safety of the filter as it previously returned None if the algorithm was unknown. Some modules, notably the user module, treated a password of None as a request not to set a password. If your playbook starts erroring because of this, change the hashing algorithm being used with this filter.

Porting custom scripts

No notable changes.

Networking

No notable changes.