Ansible 2.8 Porting Guide

This section discusses the behavioral changes between Ansible 2.7 and Ansible 2.8.

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 2.8 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_2.8.rst`, **line 13**); *backlink*
>
> Unknown interpreted text role "ref".

Playbook

Distribution Facts

The information returned for the `ansible_distribution_*` group of facts may have changed slightly. Ansible 2.8 uses a new backend library for information about distributions: nir0s/distro. This library runs on Python-3.8 and fixes many bugs, including correcting release and version names.

The two facts used in playbooks most often, `ansible_distribution` and `ansible_distribution_major_version`, should not change. If you discover a change in these facts, please file a bug so we can address the difference. However, other facts like `ansible_distribution_release` and `ansible_distribution_version` may change as erroneous information gets corrected.

Imports as handlers

Beginning in version 2.8, a task cannot notify `import_tasks` or a static `include` that is specified in `handlers`.

The goal of a static import is to act as a pre-processor, where the import is replaced by the tasks defined within the imported file. When using an import, a task can notify any of the named tasks within the imported file, but not the name of the import itself.

To achieve the results of notifying a single name but running multiple handlers, utilize `include_tasks`, or `listen` :ref:`handlers`.

> **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_2.8.rst`, **line 39**); *backlink*

Jinja Undefined values

Beginning in version 2.8, attempting to access an attribute of an Undefined value in Jinja will return another Undefined value, rather than throwing an error immediately. This means that you can now simply use a default with a value in a nested data structure when you don't know if the intermediate values are defined.

In Ansible 2.8:

```
{{ foo.bar.baz | default('DEFAULT') }}
```

In Ansible 2.7 and older:

```
{{ ((foo | default({})).bar | default({})).baz | default('DEFAULT') }}
```

or:

```
{{ foo.bar.baz if (foo is defined and foo.bar is defined and foo.bar.baz is defined) else 'DEFAULT' }}
```

Module option conversion to string

Beginning in version 2.8, Ansible will warn if a module expects a string, but a non-string value is passed and automatically converted to a string. This highlights potential problems where, for example, a `yes` or `true` (parsed as truish boolean value) would be converted to the string `'True'`, or where a version number `1.10` (parsed as float value) would be converted to `'1.1'`. Such conversions can result in unexpected behavior depending on context.

This behavior can be changed to be an error or to be ignored by setting the `ANSIBLE_STRING_CONVERSION_ACTION` environment variable, or by setting the `string_conversion_action` configuration in the `defaults` section of `ansible.cfg`.

Command line facts

`cmdline` facts returned in system will be deprecated in favor of `proc_cmdline`. This change handles special case where Kernel command line parameter contains multiple values with the same key.

Bare variables in conditionals

In Ansible 2.7 and earlier, top-level variables sometimes treated boolean strings as if they were boolean values. This led to inconsistent behavior in conditional tests built on top-level variables defined as strings. Ansible 2.8 began changing this behavior. For example, if you set two conditions like this:

```
tasks:
  - include_tasks: teardown.yml
    when: teardown

  - include_tasks: provision.yml
    when: not teardown
```

based on a variable you define **as a string** (with quotation marks around it):

- In Ansible 2.7 and earlier, the two conditions above evaluated as `True` and `False` respectively if `teardown: 'true'`
- In Ansible 2.7 and earlier, both conditions evaluated as `False` if `teardown: 'false'`
- In Ansible 2.8 and later, you have the option of disabling conditional bare variables, so `when: teardown` always evaluates as `True` and `when: not teardown` always evaluates as `False` when `teardown` is a non-empty string (including `'true'` or `'false'`)

Ultimately, `when: 'string'` will always evaluate as `True` and `when: not 'string'` will always evaluate as `False`, as long as `'string'` is not empty, even if the value of `'string'` itself looks like a boolean. For users with playbooks that depend on the old behavior, we added a config setting that preserves it. You can use the `ANSIBLE_CONDITIONAL_BARE_VARS` environment variable or `conditional_bare_variables` in the `defaults` section of `ansible.cfg` to select the behavior you want on your control node. The default setting is `true`, which preserves the old behavior. Set the config value or environment variable to `false` to start using the new option.

> **Note**
>
> In 2.10 the default setting for `conditional_bare_variables` will change to `false`. In 2.12 the old behavior will be deprecated.

Updating your playbooks

To prepare your playbooks for the new behavior, you must update your conditional statements so they accept only boolean values. For variables, you can use the `bool` filter to evaluate the string `'false'` as `False`:

```
vars:
```

```
  teardown: 'false'

tasks:
  - include_tasks: teardown.yml
    when: teardown | bool

  - include_tasks: provision.yml
    when: not teardown | bool
```

Alternatively, you can re-define your variables as boolean values (without quotation marks) instead of strings:

```
vars:
  teardown: false

tasks:
  - include_tasks: teardown.yml
    when: teardown

  - include_tasks: provision.yml
    when: not teardown
```

For dictionaries and lists, use the `length` filter to evaluate the presence of a dictionary or list as `True`:

```
- debug:
  when: my_list | length > 0

- debug:
  when: my_dictionary | length > 0
```

Do not use the `bool` filter with lists or dictionaries. If you use `bool` with a list or dict, Ansible will always evaluate it as `False`.

Double-interpolation

The `conditional_bare_variables` setting also affects variables set based on other variables. The old behavior unexpectedly double-interpolated those variables. For example:

```
vars:
  double_interpolated: 'bare_variable'
  bare_variable: false

tasks:
  - debug:
    when: double_interpolated
```

- In Ansible 2.7 and earlier, `when: double_interpolated` evaluated to the value of `bare_variable`, in this case, `False`. If the variable `bare_variable` is undefined, the conditional fails.
- In Ansible 2.8 and later, with bare variables disabled, Ansible evaluates `double_interpolated` as the string `'bare_variable'`, which is `True`.

To double-interpolate variable values, use curly braces:

```
vars:
  double_interpolated: "{{ other_variable }}"
  other_variable: false
```

Nested variables

The `conditional_bare_variables` setting does not affect nested variables. Any string value assigned to a subkey is already respected and not treated as a boolean. If `complex_variable['subkey']` is a non-empty string, then `when: complex_variable['subkey']` is always `True` and `when: not complex_variable['subkey']` is always `False`. If you want a string subkey like `complex_variable['subkey']` to be evaluated as a boolean, you must use the `bool` filter.

Gathering Facts

In Ansible 2.8 the implicit "Gathering Facts" task in a play was changed to obey play tags. Previous to 2.8, the "Gathering Facts" task would ignore play tags and tags supplied from the command line and always run in a task.

The behavior change affects the following example play.

```
- name: Configure Webservers
  hosts: webserver
  tags:
    - webserver
  tasks:
    - name: Install nginx
      package:
        name: nginx
        tags:
```

```
      - nginx
```

In Ansible 2.8, if you supply `--tags nginx`, the implicit "Gathering Facts" task will be skipped, as the task now inherits the tag of `webserver` instead of `always`.

If no play level tags are set, the "Gathering Facts" task will be given a tag of `always` and will effectively match prior behavior.

You can achieve similar results to the pre-2.8 behavior, by using an explicit `gather_facts` task in your `tasks` list.

```
- name: Configure Webservers
  hosts: webserver
  gather_facts: false
  tags:
    - webserver
  tasks:
    - name: Gathering Facts
      gather_facts:
      tags:
        - always

    - name: Install nginx
      package:
        name: nginx
      tags:
        - nginx
```

Python Interpreter Discovery

In Ansible 2.7 and earlier, Ansible defaulted to :command:`/usr/bin/python` as the setting for `ansible_python_interpreter`. If you ran Ansible against a system that installed Python with a different name or a different path, your playbooks would fail with `/usr/bin/python: bad interpreter: No such file or directory` unless you either set `ansible_python_interpreter` to the correct value for that system or added a Python interpreter and any necessary dependencies at :command:`usr/bin/python`.

> **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_2.8.rst`, **line 226);** *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_2.8.rst`, **line 226);** *backlink*
>
> Unknown interpreted text role "command".

Starting in Ansible 2.8, Ansible searches for the correct path and executable name for Python on each target system, first in a lookup table of default Python interpreters for common distros, then in an ordered fallback list of possible Python interpreter names/paths.

It's risky to rely on a Python interpreter set from the fallback list, because the interpreter may change on future runs. If an interpreter from higher in the fallback list gets installed (for example, as a side-effect of installing other packages), your original interpreter and its dependencies will no longer be used. For this reason, Ansible warns you when it uses a Python interpreter discovered from the fallback list. If you see this warning, the best solution is to explicitly set `ansible_python_interpreter` to the path of the correct interpreter for those target systems.

You can still set `ansible_python_interpreter` to a specific path at any variable level (as a host variable, in vars files, in playbooks, and so on). If you prefer to use the Python interpreter discovery behavior, use one of the four new values for `ansible_python_interpreter` introduced in Ansible 2.8:

New value	Behavior
auto |br| (future default)	If a Python interpreter is discovered, Ansible uses the discovered Python, even if :command:`/usr/bin/python` is also present. Warns when using the fallback list. **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_2.8.rst`, **line 258);** *backlink* Unknown interpreted text role "command".

New value	Behavior
	If a Python interpreter is discovered, and :command:`/usr/bin/python` is absent, Ansible uses the discovered Python. Warns when using the fallback list.
	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_2.8.rst`, **line 263**); *backlink* Unknown interpreted text role "command".
	If a Python interpreter is discovered, and :command:`/usr/bin/python` is present, Ansible uses :command:`/usr/bin/python` and prints a deprecation warning about future default behavior. Warns when using the fallback list.
auto_legacy |br| (Ansible 2.8 default)	**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_2.8.rst`, **line 268**); *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_2.8.rst`, **line 268**); *backlink* Unknown interpreted text role "command".
auto_legacy_silent	Behaves like `auto_legacy` but suppresses the deprecation and fallback-list warnings.
auto_silent	Behaves like `auto` but suppresses the fallback-list warning.

In Ansible 2.12, Ansible will switch the default from `auto_legacy` to `auto`. The difference in behaviour is that `auto_legacy` uses :command:`/usr/bin/python` if present and falls back to the discovered Python when it is not present. `auto` will always use the discovered Python, regardless of whether :command:`/usr/bin/python` exists. The `auto_legacy` setting provides compatibility with previous versions of Ansible that always defaulted to :command:`/usr/bin/python`.

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_2.8.rst`, **line 282**); *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_2.8.rst`, **line 282**); *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_2.8.rst`, **line 282**); *backlink*

Unknown interpreted text role "command".

If you installed Python and dependencies (`boto`, and so on) to :command:`/usr/bin/python` as a workaround on distros with a different default Python interpreter (for example, Ubuntu 16.04+, RHEL8, Fedora 23+), you have two options:

1. Move existing dependencies over to the default Python for each platform/distribution/version.
2. Use `auto_legacy`. This setting lets Ansible find and use the workaround Python on hosts that have it, while also finding the correct default Python on newer hosts. But remember, the default will change in 4 releases.

Retry File Creation default

In Ansible 2.8, `retry_files_enabled` now defaults to `False` instead of `True`. The behavior can be modified to previous version by editing the default `ansible.cfg` file and setting the value to `True`.

Command Line

Become Prompting

Beginning in version 2.8, by default Ansible will use the word `BECOME` to prompt you for a password for elevated privileges (`sudo` privileges on Unix systems or `enable` mode on network devices):

By default in Ansible 2.8:

```
ansible-playbook --become --ask-become-pass site.yml
BECOME password:
```

If you want the prompt to display the specific `become_method` you're using, instead of the general value `BECOME`, set :ref:`AGNOSTIC_BECOME_PROMPT` to `False` in your Ansible configuration.

By default in Ansible 2.7, or with `AGNOSTIC_BECOME_PROMPT=False` in Ansible 2.8:

```
ansible-playbook --become --ask-become-pass site.yml
SUDO password:
```

Deprecated

- Setting the async directory using `ANSIBLE_ASYNC_DIR` as an task/play environment key is deprecated and will be removed in Ansible 2.12. You can achieve the same result by setting `ansible_async_dir` as a variable like:

  ```
  - name: run task with custom async directory
    command: sleep 5
    async: 10
    vars:
      ansible_async_dir: /tmp/.ansible_async
  ```

- Plugin writers who need a `FactCache` object should be aware of two deprecations:

  1. The `FactCache` class has moved from `ansible.plugins.cache.FactCache` to `ansible.vars.fact_cache.FactCache`. This is because the `FactCache` is not part of the cache plugin API and cache plugin authors should not be subclassing it. `FactCache` is still available from its old location but will issue a deprecation warning when used from there. The old location will be removed in Ansible 2.12.
  2. The `FactCache.update()` method has been converted to follow the dict API. It now takes a dictionary as its sole argument and updates itself with the dictionary's items. The previous API where `update()` took a key and a value will now issue a deprecation warning and will be removed in 2.12. If you need the old behavior switch to `FactCache.first_order_merge()` instead.

- Supporting file-backed caching via self.cache is deprecated and will be removed in Ansible 2.12. If you maintain an inventory plugin, update it to use `self._cache` as a dictionary. For implementation details, see the :ref:`developer guide on inventory plugins<inventory_plugin_caching>`.

Unknown interpreted text role "ref".

- Importing cache plugins directly is deprecated and will be removed in Ansible 2.12. Use the plugin_loader so direct options, environment variables, and other means of configuration can be reconciled using the config system rather than constants.

```
from ansible.plugins.loader import cache_loader
cache = cache_loader.get('redis', **kwargs)
```

Modules

Major changes in popular modules are detailed here

The exec wrapper that runs PowerShell modules has been changed to set `$ErrorActionPreference = "Stop"` globally. This may mean that custom modules can fail if they implicitly relied on this behavior. To get the old behavior back, add `$ErrorActionPreference = "Continue"` to the top of the module. This change was made to restore the old behavior of the EAP that was accidentally removed in a previous release and ensure that modules are more resilient to errors that may occur in execution.

- Version 2.8.14 of Ansible changed the default mode of file-based tasks to `0o600 & ~umask` when the user did not specify a `mode` parameter on file-based tasks. This was in response to a CVE report which we have reconsidered. As a result, the `mode` change has been reverted in 2.8.15, and `mode` will now default to `0o666 & ~umask` as in previous versions of Ansible.
- If you changed any tasks to specify less restrictive permissions while using 2.8.14, those changes will be unnecessary (but will do no harm) in 2.8.15.
- To avoid the issue raised in CVE-2020-1736, specify a `mode` parameter in all file-based tasks that accept it.
- `dnf` and `yum` - As of version 2.8.15, the `dnf` module (and `yum` action when it uses `dnf`) now correctly validates GPG signatures of packages (CVE-2020-14365). If you see an error such as `Failed to validate GPG signature for [package name]`, please ensure that you have imported the correct GPG key for the DNF repository and/or package you are using. One way to do this is with the `rpm_key` module. Although we discourage it, in some cases it may be necessary to disable the GPG check. This can be done by explicitly adding `disable_gpg_check: yes` in your `dnf` or `yum` task.

Modules removed

The following modules no longer exist:

- ec2_remote_facts
- azure
- cs_nic
- netscaler
- win_msi

Deprecation notices

The following modules will be removed in Ansible 2.12. Please update your playbooks accordingly.

- `foreman` use foreman-ansible-modules instead.
- `katello` use foreman-ansible-modules instead.
- `github_hooks` use :ref:`github_webhook <ansible_2_8:github_webhook_module>` and :ref:`github_webhook_facts <ansible_2_8:github_webhook_facts_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_2.8.rst`, **line 399**); *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_2.8.rst`, **line 399**); *backlink*
>
> Unknown interpreted text role "ref".

- `digital_ocean` use :ref:`digital_ocean_droplet <ansible_2_8:digital_ocean_droplet_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_2.8.rst`, **line 400**); *backlink*
>
> Unknown interpreted text role "ref".

- `gce` use :ref:`gcp_compute_instance <ansible_2_8:gcp_compute_instance_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_2.8.rst`, **line 401**); *backlink*
  >
  > Unknown interpreted text role "ref".

- `gcspanner` use :ref:`gcp_spanner_instance <ansible_2_8:gcp_spanner_instance_module>` and :ref:`gcp_spanner_database <ansible_2_8:gcp_spanner_database_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_2.8.rst`, **line 402**); *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_2.8.rst`, **line 402**); *backlink*
  >
  > Unknown interpreted text role "ref".

- `gcdns_record` use :ref:`gcp_dns_resource_record_set <ansible_2_8:gcp_dns_resource_record_set_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_2.8.rst`, **line 403**); *backlink*
  >
  > Unknown interpreted text role "ref".

- `gcdns_zone` use :ref:`gcp_dns_managed_zone <ansible_2_8:gcp_dns_managed_zone_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_2.8.rst`, **line 404**); *backlink*
  >
  > Unknown interpreted text role "ref".

- `gcp_forwarding_rule` use :ref:`gcp_compute_global_forwarding_rule <ansible_2_8:gcp_compute_global_forwarding_rule_module>` or :ref:`gcp_compute_forwarding_rule <ansible_2_8:gcp_compute_forwarding_rule_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_2.8.rst`, **line 405**); *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_2.8.rst`, **line 405**); *backlink*
  >
  > Unknown interpreted text role "ref".

- `gcp_healthcheck` use :ref:`gcp_compute_health_check <ansible_2_8:gcp_compute_health_check_module>`, :ref:`gcp_compute_http_health_check <ansible_2_8:gcp_compute_http_health_check_module>`, or :ref:`gcp_compute_https_health_check <ansible_2_8:gcp_compute_https_health_check_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_2.8.rst`, **line 406**); *backlink*
  >
  > Unknown interpreted text role "ref".

- `gcp_backend_service` use :ref:`gcp_compute_backend_service <ansible_2_8:gcp_compute_backend_service_module>` instead.

- `gcp_target_proxy` use :ref:`gcp_compute_target_http_proxy <ansible_2_8:gcp_compute_target_http_proxy_module>` instead.

- `gcp_url_map` use :ref:`gcp_compute_url_map <ansible_2_8:gcp_compute_url_map_module>` instead.

- `panos` use the Palo Alto Networks Ansible Galaxy role instead.

Noteworthy module changes

- The `foreman` and `katello` modules have been deprecated in favor of a set of modules that are broken out per entity with better idempotency in mind.

- The `foreman` and `katello` modules replacement is officially part of the Foreman Community and supported there.

- The `tower_credential` module originally required the `ssh_key_data` to be the path to a ssh_key_file. In order to work like AWX/Tower/RHAAP, `ssh_key_data` now contains the content of the file. The previous behavior can be achieved with `lookup('file', '/path/to/file')`.

- The `win_scheduled_task` module deprecated support for specifying a trigger repetition as a list and this format will be removed in Ansible 2.12. Instead specify the repetition as a dictionary value.

- The `win_feature` module has removed the deprecated `restart_needed` return value, use the standardized `reboot_required` value instead.

- The `win_package` module has removed the deprecated `restart_required` and `exit_code` return value, use the standardized `reboot_required` and `rc` value instead.

- The `win_get_url` module has removed the deprecated `win_get_url` return dictionary, contained values are returned directly.

- The `win_get_url` module has removed the deprecated `skip_certificate_validation` option, use the standardized `validate_certs` option instead.

- The `vmware_local_role_facts` module now returns a list of dicts instead of a dict of dicts for role information.

- If `docker_network` or `docker_volume` were called with `diff: yes`, `check_mode: yes` or `debug: yes`, a return value called `diff` was returned of type `list`. To enable proper diff output, this was changed to type `dict`; the original `list` is returned as `diff.differences`.

- The `na_ontap_cluster_peer` module has replaced `source_intercluster_lif` and `dest_intercluster_lif` string options with `source_intercluster_lifs` and `dest_intercluster_lifs` list options

- The `modprobe` module now detects kernel builtins. Previously, attempting to remove (with `state: absent`) a builtin kernel module succeeded without any error message because `modprobe` did not detect the module as `present`. Now, `modprobe` will fail if a kernel module is builtin and `state: absent` (with an error message from the modprobe binary like `modprobe: ERROR: Module nfs is builtin.`), and it will succeed without reporting changed if `state: present`. Any playbooks that are using `changed_when: no` to mask this quirk can safely remove that workaround. To get the previous behavior when applying `state: absent` to a builtin kernel module, use `failed_when: false` or `ignore_errors: true` in your playbook.

- The `digital_ocean` module has been deprecated in favor of modules that do not require external dependencies. This allows for more flexibility and better module support.

- The `docker_container` module has deprecated the returned fact `docker_container`. The same value is available as the returned variable `container`. The returned fact will be removed in Ansible 2.12.

- The `docker_network` module has deprecated the returned fact `docker_container`. The same value is available as the returned variable `network`. The returned fact will be removed in Ansible 2.12.

- The `docker_volume` module has deprecated the returned fact `docker_container`. The same value is available as the returned variable `volume`. The returned fact will be removed in Ansible 2.12.

- The `docker_service` module was renamed to :ref:`docker_compose <ansible_2_8:docker_compose_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_2.8.rst`, line 463); *backlink*
  >
  > Unknown interpreted text role "ref".

- The renamed `docker_compose` module used to return one fact per service, named same as the service. A dictionary of these facts is returned as the regular return value `services`. The returned facts will be removed in Ansible 2.12.

- The `docker_swarm_service` module no longer sets a defaults for the following options:
  - `user`. Before, the default was `root`.
  - `update_delay`. Before, the default was `10`.
  - `update_parallelism`. Before, the default was `1`.

- `vmware_vm_facts` used to return dict of dict with virtual machine's facts. Ansible 2.8 and onwards will return list of dict with virtual machine's facts. Please see module `vmware_vm_facts` documentation for example.

- `vmware_guest_snapshot` module used to return `results`. Since Ansible 2.8 and onwards `results` is a reserved keyword, it is replaced by `snapshot_results`. Please see module `vmware_guest_snapshots` documentation for example.

- The `panos` modules have been deprecated in favor of using the Palo Alto Networks Ansible Galaxy role. Contributions to the role can be made here.

- The `ipa_user` module originally always sent `password` to FreeIPA regardless of whether the password changed. Now the module only sends `password` if `update_password` is set to `always`, which is the default.

- The `win_psexec` has deprecated the undocumented `extra_opts` module option. This will be removed in Ansible 2.10.

- The `win_nssm` module has deprecated the following options in favor of using the `win_service` module to configure the service after installing it with `win_nssm`: * `dependencies`, use `dependencies` of `win_service` instead * `start_mode`, use `start_mode` of `win_service` instead * `user`, use `username` of `win_service` instead * `password`, use `password` of `win_service` instead These options will be removed in Ansible 2.12.

- The `win_nssm` module has also deprecated the `start`, `stop`, and `restart` values of the `status` option. You should use the `win_service` module to control the running state of the service. This will be removed in Ansible 2.12.

- The `status` module option for `win_nssm` has changed its default value to `present`. Before, the default was `start`. Consequently, the service is no longer started by default after creation with `win_nssm`, and you should use the `win_service` module to start it if needed.

- The `app_parameters` module option for `win_nssm` has been deprecated; use `argument` instead. This will be removed in Ansible 2.12.

- The `app_parameters_free_form` module option for `win_nssm` has been aliased to the new `arguments` option.

- The `win_dsc` module will now validate the input options for a DSC resource. In previous versions invalid options would be ignored but are now not.

- The `openssl_pkcs12` module will now regenerate the pkcs12 file if there are differences between the file on disk and the parameters passed to the module.

Plugins

Plugins

- Ansible no longer defaults to the `paramiko` connection plugin when using macOS as the control node. Ansible will now use the `ssh` connection plugin by default on a macOS control node. Since `ssh` supports connection persistence between tasks and playbook runs, it performs better than `paramiko`. If you are using password authentication, you will need to install `sshpass` when using the `ssh` connection plugin. Or you can explicitly set the connection type to `paramiko` to maintain the pre-2.8 behavior on macOS.

- Connection plugins have been standardized to allow use of `ansible_<conn-type>_user` and `ansible_<conn-type>_password` variables. Variables such as `ansible_<conn-type>_pass` and `ansible_<conn-type>_username` are treated with lower priority than the standardized names and may be deprecated in the future. In general, the `ansible_user` and `ansible_password` vars should be used unless there is a reason to use the connection-specific variables.

- The `powershell` shell plugin now uses `async_dir` to define the async path for the results file and the default has changed to `%USERPROFILE%\.ansible_async`. To control this path now, either set the `ansible_async_dir` variable or the `async_dir` value in the `powershell` section of the config ini.

- Order of enabled inventory plugins (:ref:`INVENTORY_ENABLED`) has been updated, :ref:`auto <auto_inventory>` is now before :ref:`yaml <yaml_inventory>` and :ref:`ini <ini_inventory>`.

> **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_2.8.rst`, **line 526**); *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_2.8.rst`, **line 526**); *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_2.8.rst`, **line 526**); *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_2.8.rst`, **line 526**); *backlink*
>
> Unknown interpreted text role "ref".

- The private `_options` attribute has been removed from the `CallbackBase` class of callback plugins. If you have a third-party callback plugin which needs to access the command line arguments, use code like the following instead of trying to use `self._options`:

```
from ansible import context
[...]
tags = context.CLIARGS['tags']
```

  `context.CLIARGS` is a read-only dictionary so normal dictionary retrieval methods like `CLIARGS.get('tags')` and `CLIARGS['tags']` work as expected but you won't be able to modify the cli arguments at all.

- Play recap now counts `ignored` and `rescued` tasks as well as `ok`, `changed`, `unreachable`, `failed` and `skipped` tasks, thanks to two additional stat counters in the `default` callback plugin. Tasks that fail and have `ignore_errors: yes` set are listed as `ignored`. Tasks that fail and then execute a rescue section are listed as `rescued`. Note that `rescued` tasks are no longer counted as `failed` as in Ansible 2.7 (and earlier).

- `osx_say` callback plugin was renamed into :ref:`say <say_callback>`.

> **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_2.8.rst`, **line 544**); *backlink*
>
> Unknown interpreted text role "ref".

- Inventory plugins now support caching via cache plugins. To start using a cache plugin with your inventory see the section on

caching in the :ref:`inventory guide<using_inventory>`. To port a custom cache plugin to be compatible with inventory see :ref:`developer guide on cache plugins<developing_cache_plugins>`.

Porting custom scripts

Display class

As of Ansible 2.8, the `Display` class is now a "singleton". Instead of using `__main__.display` each file should import and instantiate `ansible.utils.display.Display` on its own.

OLD In Ansible 2.7 (and earlier) the following was used to access the `display` object:

```
try:
    from __main__ import display
except ImportError:
    from ansible.utils.display import Display
    display = Display()
```

NEW In Ansible 2.8 the following should be used:

```
from ansible.utils.display import Display
display = Display()
```

Networking

- The `eos_config`, `ios_config`, and `nxos_config` modules have removed the deprecated `save` and `force` parameters, use the `save_when` parameter to replicate their functionality.
- The `nxos_vrf_af` module has removed the `safi` parameter. This parameter was deprecated in Ansible 2.4 and has had no impact on the module since then.

Docutils System Messages