



redhat.

ANSIBLE

Ansible v2.0 and Beyond

About Me...



David Federlein
Director, Customer Success at Ansible

Technical Debt

Rapid organic growth for 3+ years

Bolted on features

Increasingly difficult to fix bugs

Increasingly difficult to add new features

Difficult to unit test

New Features in v2

Blocks

Grouping of related tasks

Attributes like `become`, `when`, `tags` and others can be set on a block and are then inherited by all the contained tasks

Provides a method for catching and handling errors during task execution such as roll backs

```
- block:
  - name: install Apache (Red Hat)
    yum:
      name: httpd
      state: present
  - name: create sites directories
    file:
      path: "{{ item }}"
      state: directory
    with_items: apache_dirs
  - name: copy httpd.conf
    template:
      src: httpd.conf.j2
      dest: "{{ apache_config }}"
    notify: restart apache
when: ansible_os_family == "RedHat"
tags: package
```

```
- block:
  - block:
    - name: install (Debian)
      apt:
        name: apache2
        state: present
        update_cache: yes
        cache_valid_time: 3600
      when: ansible_os_family == "Debian"
  - block:
    - name: install (Red Hat)
      yum:
        name: httpd
        state: present
      when: ansible_os_family == "RedHat"
tags: package
```

```
- hosts: localhost
  vars:
    example1: lelo
  tasks:
    - block:
      - debug: var=example1
      - debug: var=example2
    vars:
      example2: lola
    - debug: var=example2
```

```
PLAY *****

TASK [debug] *****
ok: [localhost] => {
  "example1": "lelo"
}

TASK [debug] *****
ok: [localhost] => {
  "example2": "lola"
}

TASK [debug] *****
ok: [localhost] => {
  "example2": "VARIABLE IS NOT
  DEFINED!"
}
```



```
---
- hosts: web
  tasks:
    - block:
      - debug: msg="Hello World"
      - command: /bin/false
    rescue:
      - debug: msg="I caught an error"
      - command: /bin/false
      when: ansible_os_family == "Debian"
      - debug: msg="I handled an error"
    always:
      - debug: msg="This always executes"
```

```
# any_errors_fatal on blocks works with
# 2.0.1+ (to be released)
---
- hosts: web
  tasks:
    - block:
      - deploy task 1 ...
      - deploy task 2 ...
    rescue:
      - undo task ...
      - undo task ...
    any_errors_fatal: true
```

Improved Error Messages

Playbook errors not related to syntax will show the file along with the line and column where the error occurred.

```
ERROR! Syntax Error while loading YAML.
```

```
The error appears to have been in '/path/to/test.yml': line 6, column 15, but may be elsewhere in the file depending on the exact syntax problem.
```

```
The offending line appears to be:
```

```
- debug:
  msg: {{ ansible_default_ipv4.address }}
      ^ here
```

```
We could be wrong, but this one looks like it might be an issue with missing quotes. Always quote template expression brackets when they start a value. For instance:
```

```
with_items:
  - {{ foo }}
```

```
Should be written as:
```

```
with_items:
  - "{{ foo }}"
```

Execution Strategy Plugins

Allows changes in the way tasks are executed in a play

Shipping Plugins:

- **linear** - traditional Ansible. Wait for all hosts to complete a task before continuing
- **free** - allows each host to process tasks as fast a possible without waiting for other hosts

What execution strategy will you conceive? Contribute a plugin.

```
---
- hosts: web
  strategy: free
  tasks:
    - debug:
      msg: "{{ inventory_hostname }} is starting."
    - name: "Sleep?"
      command: sleep 10
      when: ansible_os_family == "Debian"
    - debug:
      msg: "{{ inventory_hostname }} is complete."
```

Execution-Time Evaluation of Includes

Previously task include statements were pre-processed and evaluated before any tasks execution begun

Loops, facts and variables set during execution time could not be used with includes -- **now they can**

```
# This would fail before v2

- include: users.yml
  vars:
    user: "{{ item }}"
  with_items:
    - fred
    - timmy
    - alice
```


Before v2

- include: RedHat.yml
 when: ansible_os_family == "RedHat"
- include: Debian.yml
 when: ansible_os_family == "Debian"

With v2

- include: "{{ ansible_os_family }}".yml

Improved Variable Management

Centralized processing and management of all variables from all sources

Predictable order and avoids premature flattening of data structures

One shot variable resolution, instead of piecemeal as before.

Variable Precedence v1.*

1. extra vars
2. vars, vars_files, etc. aka “everything else in a playbook”
3. inventory vars — host_vars then group_vars
4. facts
5. role defaults

Variable Precedence v2.0

1. extra vars
2. task vars (only for the task)
3. block vars (only for tasks in block)
4. role and include vars
5. play vars_files
6. play vars_prompt
7. play vars
8. set_facts
9. registered vars
10. host facts
11. playbook host_vars
12. playbook group_vars
13. inventory host_vars
14. inventory group_vars
15. inventory vars
16. role defaults

Modules and Plugins

Over 200 new modules and countless improvements existing ones -- EC2, VMWare, OpenStack and Windows (still beta) amongst many others

Dozens of new inventory scripts, callbacks, lookups and other plugins

See the CHANGELOG for a complete list:

<https://github.com/ansible/ansible/blob/devel/CHANGELOG.md>

Better Use of Object Oriented Principles

More classes doing one thing

More use of inheritance and base classes especially in the plugin systems

Well defined interactions between classes

Improved ability to perform unit testing

Small Noteworthy Others

Added meta: `refresh_inventory` to force re-reading the inventory in a play. This re-executes inventory scripts, but does not force them to ignore any cache they might use.

New `delegate_facts` directive, a boolean that allows you to apply facts to the delegated host (true/yes) instead of the `inventory_hostname` (no/false) which is the default and previous behaviour.

What Will Break in v2?

Playbooks, Roles & Modules

Intended to be 100% backwards compatible for playbooks and modules -- achieved about 98.5% (mainly due to noted issues with dynamic includes)

Supporting prior unpredictable variable precedence

Support of ugly idiomatic task declarations that deserves to break like...

```
with_items: foo # is foo a variable or a string???
```

Playbooks, Roles & Modules

Empty variables and variables set to null in YAML will no longer be converted to empty strings

Template code now retains types for booleans and numbers instead of turning them into strings.

Minor change in YAML trailing line handling

Porting Guide (includes workarounds for playbooks):

http://docs.ansible.com/ansible/porting_guide_2.0.html

Problems Introduced by Dynamic Includes

Because includes are now evaluated at execution time, we can't know about:

- Tags contained in included files
 - Means `--list-tags` won't include all possible tags
 - We do not currently raise errors due to undefined tags because of this
- Handlers contained in included files
 - As with tags, we do not currently raise errors due to undefined handler notifications because of this

Problems Introduced by Dynamic Includes

Loops inside included files using a loop:

```
# include.yml

- command: echo "{{foo}} {{item}}"
  with_items: [1, 2, 3]

# test.yml
- hosts: all
  tasks:
  - include: include.yml foo={{item}}
    with_items: ['a', 'b', 'c']
```

```
changed: [localhost] => (item=1) => {"changed": true,
"cmd": ["echo", "1 1"], ...
changed: [localhost] => (item=2) => {"changed": true,
"cmd": ["echo", "2 2"], ...
changed: [localhost] => (item=3) => {"changed": true,
"cmd": ["echo", "3 3"], ...
```

```
# include.yml
- set_fact: foo="{{foo}}"
- command: echo "{{foo}} {{item}}"
  with_items: [1, 2, 3]
```

```
# test.yml
- hosts: all
  tasks:
  - include: include.yml foo={{item}}
    with_items: ['a', 'b', 'c']
```

```
changed: [localhost] => (item=1) => {"changed": true,
"cmd": ["echo", "a 1"], ...
changed: [localhost] => (item=2) => {"changed": true,
"cmd": ["echo", "a 2"], ...
changed: [localhost] => (item=3) => {"changed": true,
"cmd": ["echo", "a 3"], ...
```

Internal APIs

Callback, connection, cache and lookup plugin APIs have changed and will require modification to existing plugins

Integrating directly with Ansible's API (not plugins) will encounter breaking changes

Callbacks need to be white-listed in `ansible.cfg` -- being in the callback plugins path is not enough as in previous versions

v2.1 & Beyond

Highlights

Continued Windows infrastructure improvements

Network automation support

Continued module expansion and enhancements such as Azure and Docker

Core modules repo merged back into the main repo

Extra modules to become a separate package to install

Questions?

Thanks

ansible.com/community

ansible.com/get-started