

Ansible set_fact & add_host

Julien Girardin

March 29, 2016



A N S I B L E

Versions disclaimer

Should work on ansible ≥ 1.9
(Special mentions for ansible 2.0 only)

- 1 Server scope variable
- 2 Global scope variable
- 3 Fun facts (short)

Example

Example featuring *set_fact*

```
- name: "A task with set_fact"
  set_fact: ga=foo
  args:
    bu: bar
    zo: baz|bool
    meu: >- # add "-" to strip the \n at the end
          {{ complex_computation_in_jinja|list }}
```

site_1.yml

```
- hosts: graphite
  roles:
```

```
    - role: "graphite"
```

```
- hosts: shinken
  roles:
```

```
    - role: "shinken"
```

```
- hosts: graphite:shinken
  roles:
```

```
    - role: "ferm_basic"
```

```
- name: "Expose graphite ports"
  set_fact:
    open_port: "{{open_port|default([])
    ↪ + [2003, 80]}}"
```

```
- name: "Open tcp port in firewall"
  lineinfile:
    dest: /etc/ferm/ferm.conf
    regexp: "^@def \[ $TCP_PORTS = \["
    line: "@def $TCP_PORTS =
    ↪ ({open_port|unique|join(' ')});"
```

site_2.yml

```
- hosts: monitoring
  roles:
    - role: "graphite"

- hosts: monitoring
  roles:
    - role: "ferm_basic"

- hosts: monitoring
  roles:
    - role: "shinken"

- hosts: monitoring
  roles:
    - role: "ferm_basic"
```

TASK [graphite | Expose graphite ports] *****

ok: [graphite]

ok: [shinken]

TASK [ferm_basic | Open tcp port in firewall] *

changed: [graphite]

changed: [shinken]

TASK [shinken | Export Shinken web UI ports] **

ok: [graphite]

ok: [shinken]

TASK [ferm_basic | Open tcp port in firewall] *

changed: [graphite]

changed: [shinken]

site_2.yml

```
- hosts: monitoring
  roles:
    - role: "graphite"

- hosts: monitoring
  roles:
    - role: "ferm_basic"

- hosts: monitoring
  roles:
    - role: "shinken"

- hosts: monitoring
  roles:
    - role: "ferm_basic"
```

TASK [graphite | Expose graphite ports] *****

ok: [graphite]

ok: [shinken]

TASK [ferm_basic | Open tcp port in firewall] *

changed: [graphite]

changed: [shinken]

TASK [shinken | Export Shinken web UI ports] **

ok: [graphite]

ok: [shinken]

TASK [ferm_basic | Open tcp port in firewall] *

changed: [graphite]

changed: [shinken]

⇒ First: remove the shinken ports. Then: add it again.

site_3.yml

```
- hosts: monitoring
  roles:
```

```
  - role: "graphite"
```

```
- hosts: monitoring
  roles:
```

```
  - role: "ferm_full"
```

```
- hosts: monitoring
  roles:
```

```
  - role: "shinken"
```

```
- hosts: monitoring
  roles:
```

```
  - role: "ferm_full"
```

```
- name: "Expose graphite ports"
  set_fact:
    ports_dict: >-
      {%- set _ = ports_dict.update(
        { 'graphite.conf': [2003, 80]}) -%}
      {{ports_dict}}
# ansible 2: ports_dict
↪ |combine({'graphite.conf': [2003,
↪ 80]})
```

```
- name: "Open ports base on ports_dict var"
  copy:
    content: |-
      mod conntrack ctstate NEW ACCEPT {
        proto tcp dport (
          {{- item.value|join( " " ) }});
      }
    dest: /etc/ferm/input.d/{{ item.key }}
  with_dict: ports_dict
```


site_4.yml

- hosts: monitoring
roles:
 - role: "graphite"
- hosts: monitoring
roles:
 - role: "ferm_full"
is_last: False
- hosts: monitoring
roles:
 - role: "shinken"
- hosts: monitoring
roles:
 - role: "ferm_full"

```
- name: "Register files in input"
  set_fact:
    ferm_input: >-
      {{ ferm_input + ports_dict.keys() }}

- include: 'cleanup_input.yml'
  when: is_last|bool == True
```

cleanup_input.yml:

```
- name: "List all ferm rules"
  command: ls -A1 /etc/ferm/input.d
  register: input_rules
  failed_when: input_rules.rc not in [0, 2]
  always_run: True
  changed_when: False

- name: "/!\ Remove /\! unsued ferm rules"
  file:
    dest: "/etc/ferm/input.d/{{item}}'"
    state: absent
  with_items: >-
    {{ input_rules.stdout_lines
    ↪ |difference(ferm_input) }}
```

Warning

A missing tag *ferm* or *firewall* may un-expose a port and remove it from firewall

- 1 Server scope variable
- 2 Global scope variable
- 3 Fun facts (short)

The classical but usefull *hostvars*

```
- hosts: master
  tasks:
    - name: "Get content from master"
      slurp:
        path: /root/.ssh/id_rsa.pub
      register: sshkey
      delegate_to: "{{ groups.master[0] }}"
      run_once: True

- hosts: slaves
  tasks:
    - name: "Add sshpub key from the master"
      authorized_keys:
        key: "{{ hostvars[groups.master[0]].sshkey.content|b64decode }}"
        user: root
```

Global vars

How to pass a variable globally ?

(ie: whatever the server who sets the var and whatever the server who reads the var)

Global vars

How to pass a variable globally ?

(ie: whatever the server who sets the var and whatever the server who reads the var)

⇒ *Groups*

Example

Example featuring *add_host*

```
- name: "Example with add_hosts"
  add_host:
    name: "{{ groups.slaves[0] }}"
    groups: leader
    foo: bar
```

Common pitfall

add_host literally breaks the parallelism of ansible.
It seems to have: `run_once: True` always enabled.

Doesn't work as expected

```
- name: "Add hosts to shinken"
  add_host:
    name: "{{ inventory_hostname }}"
    groups: shinken
```

Thing you might want

```
- name: "Add hosts to shinken"
  add_host:
    name: "{{ item }}"
    groups: shinken
  with_items: "{{ play_hosts }}"
```


site_5.yml

```
- hosts: graphite
  roles:
```

```
- role: "graphite"
```

```
- hosts: shinken
  roles:
```

```
# - role: "shinken_common"
```

```
- role: "shinken_graphite"
```

```
- role: "shinken"
```

```
- hosts: graphite:shinken
  roles:
  - role: "ferm_full"
```

```
- name: "Expose graphite entry-point"
  add_host:
    name: "{{ item }}"
    groups: _graphite_entry_point
    graphite_port: 2003
  with_items: "{{ play_hosts }}"
```

shinken_graphite/templates/graphite_module.cfg.j2

```
{%- set graphite =
↳ groups._graphite_entry_point[0]
↳ -%}
define module {
    module_name      graphite
    module_type      graphite_perfdata
    host              {{ graphite }}
    port              {{
↳ hostvars[graphite].graphite_port
↳ }}
}
```

- 1 Server scope variable
- 2 Global scope variable
- 3 Fun facts (short)

Set var on another host

Since ansible ~ 1.9

```
- name:
  add_host:
    name: "another.server"
    group: all
    foo: bar
```

Variable pointer

Since ansible 2.0

```
- hosts: localhost
vars:
  key: foo
  value: ['a', 'list',]
tasks:
  - set_fact:
      '{{ key }}': '{{value}}'
  - debug: var=foo
```