목차

Ansible Summary

Patterns

- all (or *)
- hostname: foo.example.com
- groupname: webservers
- or: webservers:dbserver
- exclude: webserver:!phoenix
- intersection: webservers:&staging

패턴은 변수를 대체할 수 있음: `{{foo}}, wildcards: *.example.com,` 정규식도 포함됨: `~(web|db).*\.example\.com`

Inventory files

`'INI-file'` structure, blocks define groups. Hosts alowed in more than one group. Non-standard SSH port can follow hostname seperated by `':'` (but see also ansible_ssh_port below).

Hostname ranges: `www[01:50].example.com, db-[a:f].example.com`

Per-host variables: `foo.example.com foo=bar baz=wibble`

Speial sections:

- `[foo:children]`: new group foo contaning all members if included groups
- `[foo:vars]`: variable definitions for all members of group foo

Variable files:

YAML; given inventory file at **…**`/hosts`:

- **…**`group_vars/foo`: variable definitions for all members of group foo
- **…**`host_vars/foo.example.com`: variable definitions for foo.example.com

Behavorial inventory parameters:

- ansible_ssh_host
- ansible_ssh_port
- ansible_ssh_user
- ansible_ssh_pass
- ansible_sudo_pass
- ansible_connection
- ansible_ssh_private_key_file

- ansible_python_interpreter
- ansible_*_interpreter

Playbook example structure

Playbooks are a YAML list of one or more plays. Plays look like this - most (all?) keys are optional:

```
---
- include: playbook.yml
- name: example
  hosts: webservers
  gather_facts: no
  vars:
    http_port: 80
  vars_file:
    - "vars.yml"
    - [ "try-first.yml", "try-second-.yml" ]
  vars_prompt:
    - name: "my_password2"
      prompt: "Enter password2"
      default: "secret"
      private: yes
      encrypt: "md5_crypt"
      confirm: yes
      salt_size: 7
  remote_user: root
  sudo: yes
  sudo_user: postgress
  pre_tasks:
    - shell: echo 'pre'
    - <more tasks>
  roles:
    - common
    - { role: foo, dir: '/opt/a', port: 5000, when: "bar == 'Baz'" }
    - <more roles>
  tasks:
    - include: tasks.yml foo=bar baz=wibble
    - include: other-tasks.yml
    - name: example task
      vars:
        foo: bar
        baz:
          - one
          - two
      template: src=template.j2 dest=/etc/foo.conf
      when: ansible_os_family == "Debian"
      register: var
      notify:
```

```
      - restart apache
    remote_user: apache
    sudo: yes
    ignore_errors: True
  - <more tasks>
post_tasks:
  - shell: echo 'post'
  - <more tasks>
handlers:
  - include: handlers.yml
  - name: restart apache
    service: name=httpd state=restarted
  - <more handlers>
- <more plays>
```

Modules

List all installed modules with

```
ansible-doc --list
```

Document a particular module with

```
ansible-doc <module>
```

Show playbook snippet for specified module

```
ansible-doc -i <module>
```

Variables

Names: letters, digits, underscores; starting with a letter.

Substitution examples:

- {{ var }}
- {{ var["key1"]["key2"]}}
- {{ var.key1.key2 }}
- {{ list[0] }}

Souces:

- Highest priority:
  - `--extra-vars` on the command line
- General:
  - `vars` component of a playbook
  - From files referenced by `vars_file` in a playbook
  - From included files (incl. roles)
  - Paameters passed to includes
  - Facts (see below)
  - Any `/etc/ansible/facts.d/filename.fact` on managed machines (sets variables with `ansible_local.filename. prefix)
  - `register`: in tasks
- Lower priority:
  - Inventory (set on host or group)
- Lowest priority
  - Role defaults (from defaults/main.yml)

Built-in:

- `hostvars` (e.g. hostvars[other.example.com][...])
- `group_names` (groups containing current host)
- `groups` (all groups and hosts in the inventory)
- `inventory_hostname` (current host as in inventory)
- `inventory_hostname_short` (first component of inventory_hostname)
- `play_hosts` (hostnames in scope for current play)
- `inventory_dir` (location of the inventory)
- `inventoty_file` (name of the inventory)

Facts:

Run `ansible hostname -m setup`, but in particular:

- ansible_distribution
- ansible_distribution_release
- ansible_distribution_version
- ansible_fqdn
- ansible_hostname
- ansible_os_family
- ansible_pkg_mgr
- ansible_default_ipv4.address
- ansible_default_ipv6.address

Content of 'registered' variables:

- .rc
- .stdout

- .stdout_lines
- .changed
- .msg
- .results (when used in a loop)

Additionally available in templates:

- ansible_managed: string containing the information below
- template_host: node name of the templateâx128x153s machine
- template_uid: the owner
- template_path: absolute path of the template
- template_fullpath: the absolute path of the template
- template_run_date: the date that the template was rendered

Filters

- {{ var | to_nice_json }}
- {{ var | to_json }}
- {{ var | from_json }}
- {{ var | to_nice_yml }}
- {{ var | to_yml }}
- {{ var | from_yml }}
- {{ result | failed }}
- {{ result | changed }}
- {{ result | success }}
- {{ result | skipped }}
- {{ var | manditory }}
- {{ var | default(5) }}
- {{ list1 | unique }}
- {{ list1 | union(list2) }}
- {{ list1 | intersect(list2) }}
- {{ list1 | difference(list2) }}
- {{ list1 | symmetric_difference(list2) }}
- {{ path | basename }}
- {{ path | dirname }}
- {{ path | expanduser }}
- {{ path | realpath }}
- {{ var | b64decode }}
- {{ var | b64encode }}
- {{ filename | md5 }}
- {{ var | bool }}
- {{ var | int }}
- {{ var | quote }}
- {{ var | md5 }}
- {{ var | fileglob }}
- {{ var | match }}
- {{ var | search }}
- {{ var | regex }}

See also default jinja2 filters. In YAML, values starting { must be quoted.

Lookups

- lookup_file_etc_foo.txt
- lookup_password_tmp_passwordfile_chars_ascii
- lookup_env_home
- lookup_pipe_date
- 6379_somekey
- lookup_dnstxt_example.com
- lookup_template_._some_template.j2

Conditions

when:

- var == “Vaue”, var >= 5, etc.
- var, where var coreces to boolean (yes, true, True, TRUE)
- var is defined, var is not defined
- <condition1> and <condition2> (also or?)

Combined with `with_items`, the when statement is processed for each item. when can also be applied to includes and roles.

Loops

Standard:

```
- user: name={{ item }} state=present groups=wheel
  with_items:
    - testuser1
    - testuser2
  with_items:
    - { name: 'testuser1', groups: 'wheel' }
    - { name: 'testuser2', groups: 'root' }

  with_items: somelist
```

Nested:

```


- mysql_user: name={{ item[0] }} priv={{ item[1] }}.*:ALL
                           append_privs=yes password=foo
  with_nested:
```

```
    - [ 'alice', 'bob', 'eve' ]
    - [ 'clientdb', 'employeedb', 'providerdb' ]
```

Over hashes:

Given

```
---
users:
  alice:
    name: Alice Appleworth
    telephone: 123-456-7890
  bob:
    name: Bob Bananarama
    telephone: 987-654-3210
tasks:
  - name: Print phone records
    debug: msg="User {{ item.key }} is {{ item.value.name }}
                    ({{ item.value.telephone }})"
    with_dict: users
```

Fileglob:

```
- copy: src={{ item }} dest=/etc/fooapp/ owner=root mode=600
  with_fileglob:
    - /playbooks/files/fooapp/*
```

With content of file:

(see example for authorized_key module)

```
- authorized_key: user=deploy key="{{ item }}"
  with_file:
    - public_keys/doe-jane
    - public_keys/doe-john
```

Parallel stes of data:

Given

```
---
alpha: [ 'a', 'b', 'c', 'd' ]
numbers:  [ 1, 2, 3, 4 ]
```

```
- debug: msg="{{ item.0 }} and {{ item.1 }}"
  with_together:
    - alpha
    - numbers
```

Subelements:

Given

```
---
users:
  - name: alice
    authorized:
      - /tmp/alice/onekey.pub
      - /tmp/alice/twokey.pub
  - name: bob
    authorized:
      - /tmp/bob/id_rsa.pub

- authorized_key: "user={{ item.0.name }}
                   key='{{ lookup('file', item.1) }}'"
  with_subelements:
     - users
     - authorized
```

Integer sequence:

Decimal, hexadecimal (0x3f8) or octal (0600)

```
- user: name={{ item }} state=present groups=evens
  with_sequence: start=0 end=32 format=testuser%02x
  with_sequence: start=4 end=16 stride=2
  with_sequence: count=4
```

Random choice:

```
- debug: msg={{ item }}
  with_random_choice:
     - "go through the door"
     - "drink from the goblet"
     - "press the red button"
     - "do nothing"
```

Do-Until:

```
- action: shell /usr/bin/foo
  register: result
  until: result.stdout.find("all systems go") != -1
  retries: 5
  delay: 10
```

Results of a local program:

```
- name: Example of looping over a command result
  shell: /usr/bin/frobnicate {{ item }}
  with_lines: /usr/bin/frobnications_per_host
                       --param {{ inventory_hostname }}
```

Indexed list:

```
- name: indexed loop demo
  debug: msg="at array position {{ item.0 }} there is
                                      a value {{ item.1 }}"
  with_indexed_items: some_list
```

Flattened list:

```
---
# file: roles/foo/vars/main.yml
packages_base:
  - [ 'foo-package', 'bar-package' ]
packages_apps:
  - [ ['one-package', 'two-package' ]]
  - [ ['red-package'], ['blue-package']]
- name: flattened loop demo
  yum: name={{ item }} state=installed
  with_flattened:
    - packages_base
    - packages_apps
```

First found:

- name: template a file

```
template: src={{ item }} dest=/etc/myapp/foo.conf
with_first_found:
  - files:
```

```
      - {{ ansible_distribution }}.conf
      - default.conf
    paths:
       - search_location_one/somedir/
       - /opt/other_location/somedir/
```

</code> </WRAP>

Roles

Directory structure:

```
playbook.yml
roles/
   common/
     tasks/
       main.yml
     handlers/
       main.yml
     vars/
       main.yml
     meta/
       main.yml
     defaults/
       main.yml
     files/
     templates/
```

See documentation for role dependancies.

Tags

Both plays and tasks support a âx128x156tags:âx128x157 attribute.

```
- template: src=templates/src.j2 dest=/etc/foo.conf
  tags:
    - configuration
```

Tags can be applied to roles and includes (effectivly taggaing all included tasks)

```
roles:
    - { role: webserver, port: 5000, tags: [ 'web', 'foo' ] }
```

```
- include: foo.yml tags=web,foo
```

To select by tag:

```
ansible-playbook example.yml --tags "configuration,packages"
ansible-playbook example.yml --skip-tags "notification"
```

Command lines

ansible

```
Usage: ansible <host-pattern> [options]

Options:
  -a MODULE_ARGS, --args=MODULE_ARGS
                        module arguments
  -k, --ask-pass        ask for SSH password
  --ask-su-pass         ask for su password
  -K, --ask-sudo-pass   ask for sudo password
  --ask-vault-pass      ask for vault password
  -B SECONDS, --background=SECONDS
                        run asynchronously, failing after X seconds
                        (default=N/A)
  -C, --check           don't make any changes; instead, try to predict some
                        of the changes that may occur
  -c CONNECTION, --connection=CONNECTION
                        connection type to use (default=smart)
  -f FORKS, --forks=FORKS
                        specify number of parallel processes to use
                        (default=5)
  -h, --help            show this help message and exit
  -i INVENTORY, --inventory-file=INVENTORY
                        specify inventory host file
                        (default=/etc/ansible/hosts)
  -l SUBSET, --limit=SUBSET
                        further limit selected hosts to an additional
pattern
  --list-hosts          outputs a list of matching hosts; does not execute
                        anything else
  -m MODULE_NAME, --module-name=MODULE_NAME
                        module name to execute (default=command)
  -M MODULE_PATH, --module-path=MODULE_PATH
                        specify path(s) to module library
                        (default=/usr/share/ansible)
  -o, --one-line        condense output
  -P POLL_INTERVAL, --poll=POLL_INTERVAL
                        set the poll interval if using -B (default=15)
```

```
  --private-key=PRIVATE_KEY_FILE
                        use this file to authenticate the connection
  -S, --su              run operations with su
  -R SU_USER, --su-user=SU_USER
                        run operations with su as this user (default=root)
  -s, --sudo            run operations with sudo (nopasswd)
  -U SUDO_USER, --sudo-user=SUDO_USER
                        desired sudo user (default=root)
  -T TIMEOUT, --timeout=TIMEOUT
                        override the SSH timeout in seconds (default=10)
  -t TREE, --tree=TREE  log output to this directory
  -u REMOTE_USER, --user=REMOTE_USER
                        connect as this user (default=jw35)
  --vault-password-file=VAULT_PASSWORD_FILE
                        vault password file
  -v, --verbose         verbose mode (-vvv for more, -vvvv to enable
                        connection debugging)
  --version             show program's version number and exit
```

ansible-playbook

```
Usage: ansible-playbook playbook.yml

Options:
  -k, --ask-pass        ask for SSH password
  --ask-su-pass         ask for su password
  -K, --ask-sudo-pass   ask for sudo password
  --ask-vault-pass      ask for vault password
  -C, --check           don't make any changes; instead, try to predict some
                        of the changes that may occur
  -c CONNECTION, --connection=CONNECTION
                        connection type to use (default=smart)
  -D, --diff            when changing (small) files and templates, show the
                        differences in those files; works great with --check
  -e EXTRA_VARS, --extra-vars=EXTRA_VARS
                        set additional variables as key=value or YAML/JSON
  -f FORKS, --forks=FORKS
                        specify number of parallel processes to use
                        (default=5)
  -h, --help            show this help message and exit
  -i INVENTORY, --inventory-file=INVENTORY
                        specify inventory host file
                        (default=/etc/ansible/hosts)
  -l SUBSET, --limit=SUBSET
                        further limit selected hosts to an additional
pattern
  --list-hosts          outputs a list of matching hosts; does not execute
                        anything else
  --list-tasks          list all tasks that would be executed
```

```
  -M MODULE_PATH, --module-path=MODULE_PATH
                        specify path(s) to module library
                        (default=/usr/share/ansible)
  --private-key=PRIVATE_KEY_FILE
                        use this file to authenticate the connection
  --skip-tags=SKIP_TAGS
                        only run plays and tasks whose tags do not match
these
                        values
  --start-at-task=START_AT
                        start the playbook at the task matching this name
  --step                one-step-at-a-time: confirm each task before running
  -S, --su              run operations with su
  -R SU_USER, --su-user=SU_USER
                        run operations with su as this user (default=root)
  -s, --sudo            run operations with sudo (nopasswd)
  -U SUDO_USER, --sudo-user=SUDO_USER
                        desired sudo user (default=root)
  --syntax-check        perform a syntax check on the playbook, but do not
                        execute it
  -t TAGS, --tags=TAGS  only run plays and tasks tagged with these values
  -T TIMEOUT, --timeout=TIMEOUT
                        override the SSH timeout in seconds (default=10)
  -u REMOTE_USER, --user=REMOTE_USER
                        connect as this user (default=jw35)
  --vault-password-file=VAULT_PASSWORD_FILE
                        vault password file
  -v, --verbose         verbose mode (-vvv for more, -vvvv to enable
                        connection debugging)
  --version             show program's version number and exit
```

ansible-vault

```
Usage: ansible-vault [create|decrypt|edit|encrypt|rekey] [--help] [options]
file_name

Options:
  -h, --help  show this help message and exit

See 'ansible-vault <command> --help' for more information on a specific
command.
```

ansible-doc

```
Usage: ansible-doc [options] [module...]

Show Ansible module documentation
```

```
Options:
  --version             show program's version number and exit
  -h, --help            show this help message and exit
  -M MODULE_PATH, --module-path=MODULE_PATH
                             Ansible modules/ directory
  -l, --list            List available modules
  -s, --snippet         Show playbook snippet for specified module(s)
  -v                    Show version number and exit
```

ansible-galaxy

```
Usage: ansible-galaxy [init|info|install|list|remove] [--help] [options] ...

Options:
  -h, --help  show this help message and exit

  See 'ansible-galaxy <command> --help' for more information on a
  specific command
```

ansible-pull

Usage: ansible-pull [options] [playbook.yml]

ansible-pull: error: URL for repository not specified, use -h for help </code> </WRAP>

참조링크

- https://gist.github.com/klen/ff85fe443735afb2410d