An Ansible summary

Jon Warbrick, July 2014, V3.2 (for Ansible 1.7)

Configuration file

intro_configuration.html

First one found from of

- Contents of `$ANSIBLE_CONFIG`
- `./ansible.cfg`
- `~/.ansible.cfg`
- `/etc/ansible/ansible.cfg`

Configuration settings can be overridden by environment variables - see constants.py in the source tree for names.

Patterns

intro_patterns.html

Used on the `ansible` command line, or in playbooks.

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

Operators can be chained: `webservers:dbservers:&staging:!phoenix`

Patterns can include variable substitutions: `{{foo}}`, wildcards: `*.example.com` or 192.168.1.*, and regular expressions: `~(web|db).*\.example\.com`

Inventory files

intro_inventory.html, intro_dynamic_inventory.html

'INI-file' structure, blocks define groups. Hosts allowed in more than one group. Non-standard SSH port can follow hostname separated 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`

- `[foo:children]`: new group `foo` containing all members if included groups

- `[foo:vars]`: variable definitions for all members of group `foo`

Inventory file defaults to `/etc/ansible/hosts`. Veritable with `-i` or in the configuration file. The 'file' can also be a dynamic inventory script. If a directory, all contained files are processed.

Variable files:

intro_inventory.html

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

`group_vars` and `host_vars` directories can also exist in the playbook directory. If both paths exist, variables in the playbook directory will be loaded second.

Behavioral inventory parameters:

intro_inventory.html

- `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`

Playbooks

playbooks_intro.html, playbooks_roles.html

Playbooks are a YAML list of one or more plays. Most (all?) keys are optional. Lines can be broken on space with continuation lines indented.

Playbooks consist of a list of one or more 'plays' and/or inclusions:

```
---
- include: playbook.yml
- <play>
- ...
```

Plays

playbooks_intro.html, playbooks_roles.html, playbooks_variables.html, playbooks_conditionals.html, playbooks_acceleration.html, playbooks_delegation.html, playbooks_prompts.html, playbooks_tags.html Forum posting Forum postinb

Plays consist of play metadata and a sequence of task and handler definitions, and roles.

```
- hosts: webservers
  remote_user: root
  sudo: yes
  sudo_user: postgress
  su: yes
  su_user: exim
  gather_facts: no
  accelerate: no
  accelerate_port: 5099
  any_errors_fatal: yes
  max_fail_percentage: 30
  connection: local
  serial: 5
  vars:
    http_port: 80
  vars_files:
    - "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: 1234
      salt_size: 8
  tags:
    - stuff
    - nonsence
  pre_tasks:
    - <task>
    - ...
  roles:
    - common
    - { role: common, port: 5000, when: "bar == 'Baz'", tags :[one, two] }
    - { role: common, when: month == 'Jan' }
    - ...
  tasks:
    - include: tasks.yaml
    - include: tasks.yaml foo=bar baz=wibble
    - include: tasks.yaml
      vars:
        foo: aaa
        baz:
          - z
          - y
    - { include: tasks.yaml, foo: zzz, baz: [a,b]}
    - include: tasks.yaml
      when: day == 'Thursday'
    - <task>
    - ...
  post_tasks:
    - <task>
    - ...
  handlers:
    - include: handlers.yml
    - <task>
    - ...
```

Using `encrypt` with `vars_prompt` requires that Passlib is installed.

In addition the source code implies the availability of the following which don't *seem* to be mentioned in the documentation: `name`, `user` (deprecated), `port`, `accelerate_ipv6`, `role_names`, and `vault_password`.

Task definitions

playbooks_intro.html, playbooks_roles.html, playbooks_async.html, playbooks_checkmode.html, playbooks_delegation.html, playbooks_environment.html, playbooks_error_handling.html, playbooks_tags.html ansible-1-5-released Forum posting Ansible examples

Each task definition is a list of items, normally including at least a name and a module invocation:

```
- name: task
  remote_user: apache
  sudo: yes
  sudo_user: postgress
  sudo_pass: wibble
  su: yes
  su_user: exim
  ignore_errors: True
  delegate_to: 127.0.0.1
  async: 45
  poll: 5
  always_run: no
  run_once: false
  meta: flush_handlers
  no_log: true
  environment: <hash>
  environment:
    var1: val1
    var2: val2
  tags:
    - stuff
    - nonsence
  <module>: src=template.j2 dest=/etc/foo.conf
  action: <module>, src=template.j2 dest=/etc/foo.conf
  action: <module>
  args:
      src=template.j2
      dest=/etc/foo.conf
  local_action: <module> /usr/bin/take_out_of_pool {{ inventory_hostname }}
  when: ansible_os_family == "Debian"
  register: result
  failed_when: "'FAILED' in result.stderr"
  changed_when: result.rc != 2
  notify:
    - restart apache
```

`delegate_to: 127.0.0.1` is implied by `local_action:`

The forms `<module>: <args>`, `action: <module> <args>`, and `local_action: <module> <args>` are mutually-exclusive.

Additional keys `when_*`, `until`, `retries` and `delay` are documented below under 'Loops'.

In addition the source code implies the availability of the following which don't *seem* to be mentioned in the documentation: `first_available_file` (deprecated), `transport`, `connection`, `any_errors_fatal`.

Roles

playbooks_roles.html

Directory structure:

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

Modules

modules.htm, modules_by_category.html

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

playbooks_roles.html, playbooks_variables.html

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

Substitution examples:

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

YAML requires an item starting with a variable substitution to be quoted.

Sources:

- 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)
    - Parameters passed to includes
    - `register:` in tasks
- Lower priority:
    - Inventory (set on host or group)
- Lower priority:
    - Facts (see below)
    - Any `/etc/ansible/facts.d/filename.fact` on managed machines (sets variables with `ansible_local.filename. prefix)
- 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:

playbooks_conditionals.html, playbooks_loops.html

Depends on module. Typically includes:

- `.rc`
- `.stdout`
- `.stdout_lines`
- `.changed`
- `.msg` (following failure)
- `.results` (when used in a loop)

See also `failed`, changed, etc filters.

When used in a loop the `result` element is a list containing all responses from the module.

Additionally available in templates:

- `ansible_managed`: string containing the information below
- `template_host`: node name of the templateâ s 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

playbooks_variables.html

- `{{ 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) }}`
- `{{ ver1 | version_compare(ver2, operator='>=', strict=True }}`
- `{{ list | random }}`
- `{{ number | random }}`
- `{{ number | random(start=1, step=10) }}`
- `{{ list | join(" ") }}`
- `{{ 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 }}`
- `{{ var | regexp_replace('from', 'to' )}}`

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

Lookups

playbooks_lookups.html

Lookups are evaluated on the control machine.

- `{{ lookup('file', '/etc/foo.txt') }}`
- `{{ lookup('password', '/tmp/passwordfile length=20 chars=ascii_letters,digits') }}`
- `{{ lookup('env','HOME') }}`
- `{{ lookup('pipe','date') }}`
- `{{ lookup('redis_kv', 'redis://localhost:6379,somekey') }}`
- `{{ lookup('dnstxt', 'example.com') }}`
- `{{ lookup('template', './some_template.j2') }}`

Lookups can be assigned to variables and will be evaluated each time the variable is used.

Lookup plugins also support loop iteration (see below).

Conditions

playbooks_conditionals.html

`when:` <condition>, where condition is:

- `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. Conditional Imports and variable substitution in file and template names can avoid the need for explicit conditionals.

Loops

playbooks_loops.html

In addition the source code implies the availability of the following which don't *seem* to be mentioned in the documentation: `csvfile`, `etcd`, `inventory_hostname`.

Standard:

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

- name: add several users
  user: name={{ item.name }} state=present groups={{ item.groups }}
  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/*
```

In a role, relative paths resolve relative to the `roles/<rolename>/files` directory.

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
```

See also the `file` lookup when the content of a file is needed.

Parallel sets 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 }}
```

To loop over the results of a remote program, use `register: result` and then `with_items: result.stdout_lines` in a subsequent task.

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/
```

Tags

Both plays and tasks support a `tags:` attribute.

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

Tags can be applied to roles and includes (effectively tagging 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

playbooks_vault.html

```
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
```