

Tutorial for ansible-playbook

- -c191213
- -u191214
- author: engel-ch@pm.me
- license: MIT

Copyright © 2019, Christian Engel <mailto:engel-ch@pm.me>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Contents

- Tutorial for ansible-playbook
 - Contents
 - File & directory structure
 - hosts file
 - Execution
 - * For all hosts in the hosts file
 - * For specific hosts
 - * For some grouped hosts
 - * Tagged Execution
 - Variables
 - * Example for host-specific variables
 - Roles
 - Main main.yml file
 - Gather Facts
 - ansible-playbook CLI-options
 - Execution parameters

File & directory structure

```
hosts
./group_vars/all
./group_vars/t_rewrite
./roles/
./roles/⟨⟨roleName⟩⟩/vars
./roles/⟨⟨roleName⟩⟩/tasks
./roles/⟨⟨roleName⟩⟩/tests
./roles/⟨⟨roleName⟩⟩/meta
./roles/⟨⟨roleName⟩⟩/README.md
./roles/⟨⟨roleName⟩⟩/defaults
./roles/⟨⟨roleName⟩⟩/files
./roles/⟨⟨roleName⟩⟩/templates
./roles/⟨⟨roleName⟩⟩/.travis.yml
./roles/⟨⟨roleName⟩⟩/handlers
./host_vars/t Rewrite-cl.yml
./host_vars/t Rewrite-srv.yml
```

1. `group_vars` contains variables for a group or all hosts.

This directory usually contains the file `all`. This file is used for all hosts. Beside of this one, files like `t_rewrite` can exist. This file would only be applied for systems in the `t_rewrite` group. This file has precedence over entries in the `all` file.

2. `host_vars` contains files for specific hosts. These files end with the suffix `.yml`
3. `roles/` contain the roles as usual. A role with all potential directories can also be created using the `ansible-galaxy` command.
4. `hosts` lists the actual hosts and defines groups of hosts

hosts file

```
[t_rewrite]
t Rewrite-cl
t Rewrite-srv

[consul]
consul1
consul2
consul3
```

Hint: ansible issued a warning if the group name `t_rewrite` was written with a dash, e.g. `t Rewrite`.

Execution

For all hosts in the hosts file

```
ansible-playbook -i hosts main.yml
```

For specific hosts

Important is the comma at the end of the host list.

```
ansible-playbook -i consul1, main.yml
```

Multiple hosts can be separated by comma too.

```
ansible-playbook -i consul1,consul2, main.yml
```

For some grouped hosts

```
ansible-playbook -i hosts --limit t_rewrite main.yml
```

Tagged Execution

This call executes the commands in the main.yml file for all hosts in the `./hosts` file.

```
ansible-playbook -i hosts --limit t_rewrite -t dbg2 main.yml
```

Variables

Variables can be set on different levels:

1. variables for the execution in `ansible.cfg` or `.ansible.cfg` or `/etc/ansible.cfg`. These variables are usually not accessible in tasks.
2. global variable in `group_vars/all`
3. host group variables in `'group_vars/<>'`
4. task variables in `roles/<<roleName>>/vars/main.yml`
5. CLI specified variables supplied using the `--extra-vars <<varName>>=<<value>>` option

Example for host-specific variables

This call executes all commands tagged with `dbg2` for all hosts specified in `./hosts`.

Here an example for host-specific variables.

```
ansible-playbook -i hosts --limit t_rewrite -t hostname2 main.yml
PLAY [test1] *****
TASK [Gathering Facts] *****
ok: [t_rewrite-cl]
ok: [t_rewrite-srv]
```

```

TASK [hostname2 : show hostname debug] *****
ok: [t-rewrite-cl] => {
    "msg": "echo t-rewrite-cl.ioee2-cloud.com"
}
ok: [t-rewrite-srv] => {
    "msg": "echo t-rewrite-srv.ioee2-cloud.com"
}

```

Roles

create new role

```
ansible-galaxy init roles/profile
```

This command is helpful, because it shows a full-blown directory tree for a role. Some, good documentation is contained in the created README.md file.

Main main.yml file

This file can contain tasks and roles:

```

---

- name: test1
  hosts: all
  become: yes
  become_method: sudo
  gather_facts: true

  tasks:
    - name: dbg
      debug:
        msg: " echo {{ message }}"

    - name: dbg2
      debug:
        msg: "echo dbg2 {{message}}"
      tags:
        - dbg2

  roles:
    - name: profile
      roles:
        - profile
      tags:
        - profile

- name: hostname2
  roles:
    - hostname2

```

```
tags:
  - hostname2
```

Gather Facts

gather facts

```
ansible all -i hosts -m gather_facts > ansible.facts
```

ansible-playbook CLI-options

```
--list-tags
--list-tasks
--list-hosts
--skip-tags .... # do not execute the specified tags
--tags ... # execute the specified tags
```

Execution parameters

They can be specified in `./ansible.cfg`. These values cannot easily be accessed in the ansible execution. Access requires either the action plugin or vars plugin.

Here is an example file:

```
[defaults]

timeout = 33
interpreter_python = /usr/bin/python3
```