
ansible-role-ansible Documentation

Release 1.10.3

Vladimir Botka

Jul 05, 2020

TABLE OF CONTENTS:

1	User's guide	3
1.1	Introduction	3
1.2	Installation	4
1.3	Playbook	4
1.4	Debug	5
1.5	Tags	5
1.6	Tasks	5
1.7	Variables	15
1.8	Default variables	15
1.9	Best practice	16
2	Annotated source code	19
2.1	Tasks	19
3	Copyright	29
4	Legal Notice	31
5	Indices and tables	33

This [role](#) and the documentation is work in progress. Please feel free to [share your feedback and report issues](#). Contributions are welcome.

USER'S GUIDE

Table of Contents

- *User's guide*
 - *Introduction*
 - *Installation*
 - *Playbook*
 - *Debug*
 - *Tags*
 - *Tasks*
 - * *Configure Ansible*
 - *Synopsis*
 - * *Ansible plugins*
 - *Synopsis*
 - * *Ansible devel*
 - *Synopsis*
 - *Variables*
 - *Default variables*
 - *Best practice*

1.1 Introduction

Run this role to install and configure Ansible. Optionally install Ansible plugins, checkout repo with the source code and download the Ansible release notes.

- Ansible role: `ansible`
- Supported systems:
 - FreeBSD Supported Production Releases
 - Ubuntu Supported Releases

- Requirements in future releases: [ansible_lib](#)

See also:

- [Ansible project documentation](#)
- [Ansible project source code](#)
- [Other Tools And Programs](#)

1.2 Installation

The most convenient way how to install an Ansible role is to use Ansible Galaxy CLI `ansible-galaxy`. The utility comes with the standard Ansible package and provides the user with a simple interface to the Ansible Galaxy's services. For example, take a look at the current status of the role

```
shell> ansible-galaxy info vbotka.ansible
```

and install it

```
shell> ansible-galaxy install vbotka.ansible
```

Install the library of tasks (for future releases)

```
shell> ansible-galaxy install vbotka.ansible_lib
```

See also:

- To install specific versions from various sources see [Installing content](#).
- Take a look at other roles `shell> ansible-galaxy search --author=vbotka`

1.3 Playbook

Below is a simple playbook that calls this role at a single host `srv.example.com` (2)

```
1 shell> cat ansible.yml
2 - hosts: srv.example.com
3   gather_facts: true
4   connection: ssh
5   remote_user: admin
6   become: yes
7   become_user: root
8   become_method: sudo
9   roles:
10    - vbotka.ansible
```

Note: `gather_facts: true` (3) must be set to gather facts needed to evaluate OS-specific options of the role. For example to install packages the variable `ansible_os_family` is needed to select the appropriate Ansible module.

See also:

- For details see [Connection Plugins](#) (4-5)

- See also [Understanding Privilege Escalation \(6-8\)](#)

1.4 Debug

To see additional debug information enable debug output in the configuration

```
lp_debug: true
```

, or set the extra variable in the command

```
shell> ansible-playbook ansible.yml -e 'lp_debug=true'
```

Note: The debug output of this role is optimized for the **yaml** callback plugin. Set this plugin for example in the environment `shell> export ANSIBLE_STDOUT_CALLBACK=yaml`.

See also:

- [Playbook Debugger](#)

1.5 Tags

The tags provide the user with a very useful tool to run selected tasks of the role. To see what tags are available list the tags of the role with the command

```
shell> ansible-playbook ansible.yml --list-tags

playbook: ansible.yml

play #1 (srv.example.com): srv.example.com TAGS: [] TASK TAGS:
  [always, ma_config, ma_debug, ma_devel, ma_devel_repo,
  ma_devel_rnotes, ma_packages, ma_plugins, ma_plugins_dirs,
  ma_plugins_download, ma_plugins_extract, ma_plugins_link,
  ma_plugins_lists, ma_plugins_test_inikeys, ma_vars]
```

For example, display the list of the variables and their values with the tag `lp_debug` (when the debug is enabled `lp_debug: true`)

```
shell> ansible-playbook ansible.yml -t lp_debug
```

See what packages will be installed

```
shell> ansible-playbook ansible.yml -t lp_packages --check
```

Install packages and exit the play

```
shell> ansible-playbook ansible.yml -t lp_packages
```

1.6 Tasks

Test single tasks at single remote host `test_01`. Create a playbook

```
shell> cat ansible.yml
- hosts: test_01
  become: true
  roles:
    - vbotka.ansible
```

Customize configuration in `host_vars/test_01/ma-*.yaml` and check the syntax

```
shell> ansible-playbook ansible.yml --syntax-check
```

Then dry-run the selected task and see what will be changed. Replace `<tag>` with valid tag.

```
shell> ansible-playbook ansible.yml -t <tag> --check --diff
```

When all seems to be ready run the command. Run the command twice and make sure the playbook and the configuration is idempotent

```
shell> ansible-playbook ansible -t <tag>
```

1.6.1 Configure Ansible

Synopsis

Configure Ansible <WIP>.

See also:

- Annotated Source code [configure.yml](#)
- [Configuring Ansible](#)

1.6.2 Ansible plugins

Synopsis

Install Ansible plugins.

See also:

- Annotated Source code [plugins.yml](#)
- [Ansible plugins](#)
- [Mitogen for Ansible](#)

Example 1: Install Ansible plugins

Create a playbook

```
shell> cat ansible.yml
- hosts: test_01
  become: true
  roles:
    - vbotka.ansible
```

Create `host_vars/test_01/ansible-plugins.yml`

```
shell> cat host_vars/test_01/ansible-plugins.yml
map_vbotka_ver: 1.7.0
map_vbotka_sha256:
↪ "sha256:c794517999ae83eea615cb65e337e8c7b9ce2636b34d75c625641f8b804bcc03"
ma_plugins:
- archive: ansible-plugins-{{ map_vbotka_ver }}.tar.gz
  archive_url: https://github.com/vbotka/ansible-plugins/archive/{{ map_vbotka_ver }}
↪ }.tar.gz
  checksum: "{{ map_vbotka_sha256 }}"
  dest: ansible-plugins-{{ map_vbotka_ver }}
  link: ansible-plugins
  plugins:
    - path: ansible-plugins/filter_plugins/dict_filters
      ini_key: filter_plugins
      enable: true
    - path: ansible-plugins/filter_plugins/hash_filters
      ini_key: filter_plugins
      enable: true
    - path: ansible-plugins/filter_plugins/list_filters
      ini_key: filter_plugins
      enable: true
```

Install plugins

```
shell> ansible-playbook ansible.yml -t ma_plugins
...
TASK [vbotka.ansible : plugins: Debug ma_plugins_paths_list] *****
ok: [planb] => (item=filter_plugins) => {
  "msg": {
    "filter_plugins": {
      "paths": [
        "ansible-plugins/filter_plugins/dict_filters",
        "ansible-plugins/filter_plugins/hash_filters",
        "ansible-plugins/filter_plugins/list_filters",
      ]
    }
  }
}
```

Show the installed plugins at the remote host

```
test_01> tree /usr/local/ansible/plugins/ansible-plugins/filter_plugins/
/usr/local/ansible/plugins/ansible-plugins/filter_plugins/
├── dict_filters
│   └── dict_filters.py
├── hash_filters
│   └── hash_filters.py
└── list_filters
    └── list_filters.py
```

Example 2: Install Mitogen plugins

Create a playbook

```
shell> cat ansible.yml
- hosts: test_01
  become: true
  roles:
    - vbotka.ansible
```

Create *host_vars/test_01/ansible-plugins.yml*

```
shell> cat host_vars/test_01/ansible-plugins.yml
map_mitogen_ver: 0.2.9
map_mitogen_sha256:
  ↪ "sha256:76cb9afef92596818a4639afb2a0bb0384ce7b6699b353af55662057b08b1e57"
ma_plugins:
  - archive: mitogen-{{ map_mitogen_ver }}.tar.gz
    archive_url: https://networkgenomics.com/try/mitogen-{{ map_mitogen_ver }}.tar.gz
    checksum: "{{ map_mitogen_sha256 }}"
    dest: mitogen-{{ map_mitogen_ver }}
    link: mitogen
    plugins:
      - path: mitogen/ansible_mitogen/plugins/strategy
        ini_key: strategy_plugins
        enable: true
```

Install plugins

```
shell> ansible-playbook ansible.yml -t ma_plugins
...
TASK [vbotka.ansible : plugins: Debug ma_plugins_paths_list] *****
ok: [planb] => (item=filter_plugins) => {
  "msg": {
    "strategy_plugins": {
      "paths": [
        "mitogen/ansible_mitogen/plugins/strategy"
      ]
    }
  }
}
```

Show the installed plugins at the remote host

```
shell> tree /usr/local/ansible/plugins/mitogen/ansible_mitogen/plugins/
/usr/local/ansible/plugins/mitogen/ansible_mitogen/plugins/
├── action
│   ├── __init__.py
│   ├── mitogen_fetch.py
│   ├── mitogen_fetch.pyc
│   └── mitogen_get_stack.py
├── connection
│   ├── __init__.py
│   ├── mitogen_buildah.py
│   ├── mitogen_doas.py
│   ├── mitogen_docker.py
│   ├── mitogen_jail.py
│   ├── mitogen_kubectl.py
│   ├── mitogen_local.py
│   ├── mitogen_local.pyc
│   └── mitogen_lxc.py
```

(continues on next page)

(continued from previous page)

```

├── mitogen_lxd.py
├── mitogen_machinectl.py
├── mitogen_setns.py
├── mitogen_ssh.py
├── mitogen_ssh.pyc
├── mitogen_sudo.py
├── mitogen_su.py
├── __init__.py
├── strategy
│   ├── __init__.py
│   ├── mitogen_free.py
│   ├── mitogen_host_pinned.py
│   ├── mitogen_linear.py
│   ├── mitogen_linear.pyc
│   └── mitogen.py

```

Example 1: Install Ansible plugins

Create a playbook

```

shell> cat ansible.yml
- hosts: test_01
  become: true
  roles:
    - vbotka.ansible

```

Create *host_vars/test_01/ansible-plugins.yml*

```

shell> cat host_vars/test_01/ansible-plugins.yml
map_vbotka_ver: 1.7.0
map_vbotka_sha256:
  ↪ "sha256:c794517999ae83eea615cb65e337e8c7b9ce2636b34d75c625641f8b804bcc03"
ma_plugins:
  - archive: ansible-plugins-{{ map_vbotka_ver }}.tar.gz
    archive_url: https://github.com/vbotka/ansible-plugins/archive/{{ map_vbotka_ver }}
    ↪}.tar.gz
    checksum: "{{ map_vbotka_sha256 }}"
    dest: ansible-plugins-{{ map_vbotka_ver }}
    link: ansible-plugins
    plugins:
      - path: ansible-plugins/filter_plugins/dict_filters
        ini_key: filter_plugins
        enable: true
      - path: ansible-plugins/filter_plugins/hash_filters
        ini_key: filter_plugins
        enable: true
      - path: ansible-plugins/filter_plugins/list_filters
        ini_key: filter_plugins
        enable: true

```

Install plugins

```

shell> ansible-playbook ansible.yml -t ma_plugins
...
TASK [vbotka.ansible : plugins: Debug ma_plugins_paths_list] *****

```

(continues on next page)

(continued from previous page)

```
ok: [planb] => (item=filter_plugins) => {
  "msg": {
    "filter_plugins": {
      "paths": [
        "ansible-plugins/filter_plugins/dict_filters",
        "ansible-plugins/filter_plugins/hash_filters",
        "ansible-plugins/filter_plugins/list_filters",
      ]
    }
  }
}
```

Show the installed plugins at the remote host

```
test_01> tree /usr/local/ansible/plugins/ansible-plugins/filter_plugins/
/usr/local/ansible/plugins/ansible-plugins/filter_plugins/
├── dict_filters
│   └── dict_filters.py
├── hash_filters
│   └── hash_filters.py
└── list_filters
    └── list_filters.py
```

Example 2: Install Mitogen plugins

Create a playbook

```
shell> cat ansible.yml
- hosts: test_01
  become: true
  roles:
    - vbotka.ansible
```

Create *host_vars/test_01/ansible-plugins.yml*

```
shell> cat host_vars/test_01/ansible-plugins.yml
map_mitogen_ver: 0.2.9
map_mitogen_sha256:
  ↪ "sha256:76cb9afef92596818a4639afb2a0bb0384ce7b6699b353af55662057b08b1e57"
ma_plugins:
  - archive: mitogen-{{ map_mitogen_ver }}.tar.gz
    archive_url: https://networkgenomics.com/try/mitogen-{{ map_mitogen_ver }}.tar.gz
    checksum: "{{ map_mitogen_sha256 }}"
    dest: mitogen-{{ map_mitogen_ver }}
    link: mitogen
    plugins:
      - path: mitogen/ansible_mitogen/plugins/strategy
        ini_key: strategy_plugins
        enable: true
```

Install plugins

```
shell> ansible-playbook ansible.yml -t ma_plugins
...
TASK [vbotka.ansible : plugins: Debug ma_plugins_paths_list] *****
```

(continues on next page)

(continued from previous page)

```
ok: [planb] => (item=filter_plugins) => {
  "msg": {
    "strategy_plugins": {
      "paths": [
        "mitogen/ansible_mitogen/plugins/strategy"
      ]
    }
  }
}
```

Show the installed plugins at the remote host

```
shell> tree /usr/local/ansible/plugins/mitogen/ansible_mitogen/plugins/
/usr/local/ansible/plugins/mitogen/ansible_mitogen/plugins/
├── action
│   ├── __init__.py
│   ├── mitogen_fetch.py
│   ├── mitogen_fetch.pyc
│   └── mitogen_get_stack.py
├── connection
│   ├── __init__.py
│   ├── mitogen_buildah.py
│   ├── mitogen_doas.py
│   ├── mitogen_docker.py
│   ├── mitogen_jail.py
│   ├── mitogen_kubectl.py
│   ├── mitogen_local.py
│   ├── mitogen_local.pyc
│   ├── mitogen_lxc.py
│   ├── mitogen_lxd.py
│   ├── mitogen_machinectl.py
│   ├── mitogen_setns.py
│   ├── mitogen_ssh.py
│   ├── mitogen_ssh.pyc
│   ├── mitogen_sudo.py
│   └── mitogen_su.py
├── __init__.py
├── strategy
│   ├── __init__.py
│   ├── mitogen_free.py
│   ├── mitogen_host_pinned.py
│   ├── mitogen_linear.py
│   ├── mitogen_linear.pyc
│   └── mitogen.py
```

1.6.3 Ansible devel

Synopsis

Ansible development <WIP>.

See also:

- Annotated Source code [devel.yml](#)
- [Ansible Developer Guide](#)

Example 1: Ansible checkout repository

Create a playbook

```
shell> cat ansible.yml
- hosts: test_01
  become: true
  roles:
    - vbotka.ansible
```

Create *host_vars/test_01/ansible.yml*

```
shell> cat host_vars/test_01/ansible.yml
ma_devel: true
ma_repo_dir: /scratch/ansible
ma_repo_version: devel
```

Checkout repository to the directory */scratch/ansible-git* and checkout branch *devel*

```
shell> > ansible-playbook ansible.yml -t ma_devel_repo
...
TASK [vbotka.ansible : devel: Create directory /scratch/ansible]_
↪*****
skipping: [test_01]

TASK [vbotka.ansible : devel: Checkout devel]_
↪*****
skipping: [test_01]
```

Show the installed repo

```
test_01> ls -l /scratch/ansible
bin
changelogs
COPYING
docs
examples
hacking
lib
licenses
Makefile
MANIFEST.in
packaging
README.rst
requirements.txt
setup.py
shippable.yml
test
```

Example 2: Ansible get release notes

Create a playbook

```
shell> cat ansible.yml
- hosts: test_01
```

(continues on next page)

(continued from previous page)

```
become: true
roles:
  - vbotka.ansible
```

Create *host_vars/test_01/ansible.yml*

```
shell> cat host_vars/test_01/ansible.yml
ma_devel: true
ma_rnotes_dir: /scratch/ansible-release-notes
ma_rnotes_list: [ '2.5', '2.6', '2.7', '2.8', '2.9', '2.10' ]
```

Get Ansible release notes to the directory */scratch/ansible-release-notes*

```
shell> ansible-playbook ansible.yml -t ma_devel_rnotes
...
TASK [vbotka.ansible : devel: Create directory /scratch/ansible-release-notes]
↳ *****
ok: [test_01]

TASK [vbotka.ansible : devel: Get release notes ['2.5', '2.6', '2.7', '2.8', '2.9',
↳ '2.10']]
ok: [test_01] => (item=2.5)
ok: [test_01] => (item=2.6)
ok: [test_01] => (item=2.7)
ok: [test_01] => (item=2.8)
ok: [test_01] => (item=2.9)
ok: [test_01] => (item=2.10)
```

Show the release notes at the remote host

```
test_01> tree /scratch/ansible-release-notes
/scratch/ansible-release-notes
├── CHANGELOG-v2.10.rst
├── CHANGELOG-v2.5.rst
├── CHANGELOG-v2.6.rst
├── CHANGELOG-v2.7.rst
├── CHANGELOG-v2.8.rst
└── CHANGELOG-v2.9.rst
```

Example 1: Ansible checkout repository

Create a playbook

```
shell> cat ansible.yml
- hosts: test_01
  become: true
  roles:
    - vbotka.ansible
```

Create *host_vars/test_01/ansible.yml*

```
shell> cat host_vars/test_01/ansible.yml
ma_devel: true
ma_repo_dir: /scratch/ansible
ma_repo_version: devel
```

Checkout repository to the directory */scratch/ansible-git* and checkout branch *devel*

```
shell> > ansible-playbook ansible.yml -t ma_devel_repo
...
TASK [vbotka.ansible : devel: Create directory /scratch/ansible]_
↪*****
skipping: [test_01]

TASK [vbotka.ansible : devel: Checkout devel]_
↪*****
skipping: [test_01]
```

Show the installed repo

```
test_01> ls -l /scratch/ansible
bin
changelogs
COPYING
docs
examples
hacking
lib
licenses
Makefile
MANIFEST.in
packaging
README.rst
requirements.txt
setup.py
shippable.yml
test
```

Example 2: Ansible get release notes

Create a playbook

```
shell> cat ansible.yml
- hosts: test_01
  become: true
  roles:
    - vbotka.ansible
```

Create *host_vars/test_01/ansible.yml*

```
shell> cat host_vars/test_01/ansible.yml
ma_devel: true
ma_rnotes_dir: /scratch/ansible-release-notes
ma_rnotes_list: [ '2.5', '2.6', '2.7', '2.8', '2.9', '2.10']
```

Get Ansible release notes to the directory */scratch/ansible-release-notes*

```
shell> ansible-playbook ansible.yml -t ma_devel_rnotes
...
TASK [vbotka.ansible : devel: Create directory /scratch/ansible-release-notes]_
↪*****
ok: [test_01]
```

(continues on next page)

(continued from previous page)

```
TASK [vbotka.ansible : devel: Get release notes ['2.5', '2.6', '2.7', '2.8', '2.9',
↪ '2.10']]
ok: [test_01] => (item=2.5)
ok: [test_01] => (item=2.6)
ok: [test_01] => (item=2.7)
ok: [test_01] => (item=2.8)
ok: [test_01] => (item=2.9)
ok: [test_01] => (item=2.10)
```

Show the release notes at the remote host

```
test_01> tree /scratch/ansible-release-notes
/scratch/ansible-release-notes
├── CHANGELOG-v2.10.rst
├── CHANGELOG-v2.5.rst
├── CHANGELOG-v2.6.rst
├── CHANGELOG-v2.7.rst
├── CHANGELOG-v2.8.rst
└── CHANGELOG-v2.9.rst
```

1.7 Variables

In this chapter we describe role's default variables stored in the directory **defaults**.

See also:

- Ansible variable precedence: Where should I put a variable?

1.8 Default variables

<TBD>

[defaults/main.yml]

```
1 ---
2 # default vars for ansible
3
4 ma_install: true
5 ma_debug: false
6 ma_backup_conf: false
7
8 # Packages
9 freebsd_install_method: packages
10 # freebsd_install_method: ports
11 freebsd_use_packages: true
12 freebsd_install_retries: 10
13 freebsd_install_delay: 5
14 linux_install_retries: 10
15 linux_install_delay: 5
16
17 # Module to configure Ansible:
18 # template    flagged:stableinterface/core
```

(continues on next page)

(continued from previous page)

```

19 # lineinfile flagged:preview/core (tested OK)
20 # ini_file   flagged:preview/community (buggy)
21 ma_config_type: template
22 ma_config_template_default: ansible-default.cfg.j2
23 ma_config: []
24
25 ma_src_path: /usr/local/ansible/src
26
27 # Plugins
28 ma_plugins_paths_list: []
29 ma_plugins_path: /usr/local/ansible/plugins
30 ma_plugins: []
31 ma_plugins_sections:
32     action_plugins: defaults
33     cache_plugins: defaults
34     callback_plugins: defaults
35     connection_plugins: defaults
36     filter_plugins: defaults
37     httpapi_plugins: defaults
38     inventory_plugins: defaults
39     lookup_plugins: defaults
40     netconf_plugins: defaults
41     strategy_plugins: defaults
42     terminal_plugins: defaults
43     test_plugins: defaults
44     vars_plugins: defaults
45
46 # Devel
47 ma_devel: false
48 ma_devel_owner: root
49 ma_repo: false
50 ma_repo_version: devel
51 ma_repo_url: https://github.com/ansible/ansible
52 ma_repo_dir: /scratch/ansible-git
53 ma_rnotes: false
54 ma_rnotes_url: 'https://raw.githubusercontent.com/ansible/ansible/stable-'
55 ma_rnotes_dir: /scratch/ansible-release-notes
56 ma_rnotes_list: [ '2.5', '2.6', '2.7', '2.8', '2.9', '2.10' ]
57
58 # EOF
59 ...

```

1.9 Best practice

Display the variables for debug if needed. Then disable this task `ma_debug: false` to speedup the playbook

```
shell> ansible-playbook ansible.yml -t ma_debug
```

Install packages Then disable this task `ma_install: false` to speedup the playbook

```
shell> ansible-playbook ansible.yml -t ma_packages \
      -e 'ma_install=true'
```

The role and the configuration data in the examples are idempotent. Once the installation and configuration have passed there should be no changes reported by *ansible-playbook* when running the playbook repeatedly. Disable

debug, and install to speedup the playbook

```
shell> ansible-playbook ansible.yml
```


ANNOTATED SOURCE CODE

Table of Contents

- *Annotated source code*
 - *Tasks*
 - * *main.yml*
 - * *configure.yml*
 - * *debug.yml*
 - * *devel.yml*
 - * *packages.yml*
 - * *plugins.yml*
 - * *vars.yml*

2.1 Tasks

2.1.1 main.yml

Synopsis: Tasks of the playbook.

Description of the task.

[main.yml]

```
1 ---
2 # tasks for ansible
3
4 - import_tasks: vars.yml
5   tags: [ma_vars, always]
6
7 - import_tasks: debug.yml
8   when: ma_debug|bool
9   tags: [ma_debug, always]
10
11 - import_tasks: packages.yml
12   when: ma_install|bool
13   tags: ma_packages
```

(continues on next page)

(continued from previous page)

```

14
15 - import_tasks: plugins.yml
16   when: ma_plugins|length > 0
17   tags: ma_plugins
18
19 - import_tasks: configure.yml
20   when: ma_config|length > 0
21   tags: ma_config
22
23 - import_tasks: devel.yml
24   when: ma_devel|bool
25   tags: ma_devel
26
27 # EOF
28 ...

```

2.1.2 configure.yml

Synopsis: Configure configure.

Description of the task.

[configure.yml]

```

1 ---
2
3 # template flagged:stableinterface/core
4 - block:
5   - name: "configure: Create directories for Ansible configuration"
6     file:
7       dest: "{{ item.path | dirname }}"
8       state: directory
9       loop: "{{ ma_config }}"
10      loop_control:
11        label: "{{ item.path }}"
12    - name: "configure: Ansible configuration from template"
13      template:
14        src: "{{ item.template|default(ma_config_template_default) }}"
15        dest: "{{ item.path }}"
16        owner: "{{ item.owner }}"
17        group: "{{ item.group }}"
18        mode: "{{ item.mode }}"
19        backup: "{{ ma_backup_conf }}"
20        loop: "{{ ma_config }}"
21        loop_control:
22          label: "{{ item.path }}"
23      when:
24        - ma_config_type|lower == "template"
25        - ma_config|length > 0
26
27 # lineinfile flagged:preview/core (tested OK)
28 - name: "configure: Ansible configuration by lineinfile"
29   lineinfile:
30     path: "{{ item.0.path }}"
31     owner: "{{ item.0.owner }}"
32     group: "{{ item.0.group }}"

```

(continues on next page)

(continued from previous page)

```

33     mode: "{{ item.0.mode }}"
34     regexp: "^{{ item.1.key }}"
35     line: "{{ item.1.key }}={{ item.1.value }}"
36     insertafter: "\\[{{ item.1.section }}\\]"
37     create: true
38     backup: "{{ ma_backup_conf }}"
39     loop: "{{ ma_config|subelements('config') }}"
40     loop_control:
41         label: "{{ item.0.path }}"
42     when:
43         - ma_config_type|lower == "lineinfile"
44         - ma_config|length > 0
45
46 # ini_file flagged:preview/community (buggy)
47 - name: "configure: Ansible configuration by ini_file"
48   ini_file:
49     path: "{{ item.0.path }}"
50     owner: "{{ item.0.owner }}"
51     group: "{{ item.0.group }}"
52     mode: "{{ item.0.mode }}"
53     section: "{{ item.1.section }}"
54     option: "{{ item.1.key }}"
55     value: "{{ item.1.key }}={{ item.1.value }}"
56     create: true
57     backup: "{{ ma_backup_conf }}"
58     loop: "{{ ma_config|subelements('config') }}"
59     loop_control:
60         label: "{{ item.0.path }}"
61     when:
62         - ma_config_type|lower == "ini_file"
63         - ma_config|length > 0
64
65 # EOF
66 ...

```

2.1.3 debug.yml

Synopsis: Configure debug.

Description of the task.

[debug.yml]

```

1 ---
2
3 - name: "Debug"
4   vars:
5     msg: |
6       ansible_architecture [{{ ansible_architecture }}]
7       ansible_os_family [{{ ansible_os_family }}]
8       ansible_distribution [{{ ansible_distribution }}]
9       ansible_distribution_major_version [{{ ansible_distribution_major_version }}]
10      ansible_distribution_version [{{ ansible_distribution_version }}]
11      ansible_distribution_release [{{ ansible_distribution_release }}]
12      ansible_python_version [{{ ansible_python_version }}]
13

```

(continues on next page)

(continued from previous page)

```

14     ma_install [{{ ma_install }}]
15     ma_backup_conf [{{ ma_backup_conf }}]
16     freebsd_install_method [{{ freebsd_install_method }}]
17     freebsd_use_packages [{{ freebsd_use_packages }}]
18     freebsd_install_retries [{{ freebsd_install_retries }}]
19     freebsd_install_delay [{{ freebsd_install_delay }}]
20     linux_install_retries [{{ linux_install_retries }}]
21     linux_install_delay [{{ linux_install_delay }}]
22     ma_packages
23     {{ ma_packages|to_nice_yaml }}
24     ma_plugins_paths_list [{{ ma_plugins_paths_list }}]
25     ma_plugins_path [{{ ma_plugins_path }}]
26     ma_src_path [{{ ma_src_path }}]
27     ma_plugins
28     {{ ma_plugins|to_yaml }}
29     ma_config_type [{{ ma_config_type }}]
30     ma_config_template_default [{{ ma_config_template_default }}]
31     ma_config
32     {{ ma_config|to_yaml }}
33     ma_devel [{{ ma_devel }}]
34     ma_devel_owner [{{ ma_devel_owner }}]
35     ma_devel_group [{{ ma_devel_group|default('UNDEFINED') }}]
36     ma_repo [{{ ma_repo }}]
37     ma_repo_url [{{ ma_repo_url }}]
38     ma_repo_version [{{ ma_repo_version }}]
39     ma_repo_dir [{{ ma_repo_dir }}]
40     ma_rnotes [{{ ma_rnotes }}]
41     ma_rnotes_list [{{ ma_rnotes_list }}]
42     ma_rnotes_url [{{ ma_rnotes_url }}]
43     ma_rnotes_dir [{{ ma_rnotes_dir }}]
44     ma_rnotes_list [{{ ma_rnotes_list }}]
45     debug:
46     msg: "{{ msg.split('\n')[:-1] }}"
47
48 # EOF
49 ...

```

2.1.4 devel.yml

Synopsis: Configure devel.

Description of the task.

[devel.yml]

```

1 ---
2
3 - name: "devel: Checkout repository"
4   block:
5     - name: "devel: Create directory {{ ma_repo_dir }}"
6       file:
7         state: directory
8         path: "{{ ma_repo_dir }}"
9         owner: "{{ ma_devel_owner }}"
10        group: "{{ ma_devel_group|default(omit) }}"
11    - name: "devel: Checkout {{ ma_repo_version }}"

```

(continues on next page)

(continued from previous page)

```

12     become: "{{ ma_devel_owner }}"
13     git:
14         repo: "{{ ma_repo_url }}"
15         version: "{{ ma_repo_version }}"
16         dest: "{{ ma_repo_dir }}"
17     when: ma_repo|bool
18     tags: ma_devel_repo
19
20 - name: "devel: Get release notes"
21   block:
22     - name: "devel: Create directory {{ ma_rnotes_dir }}"
23       file:
24         state: directory
25         path: "{{ ma_rnotes_dir }}"
26         owner: "{{ ma_devel_owner }}"
27         group: "{{ ma_devel_group|default(omit) }}"
28     - name: "devel: Get release notes {{ ma_rnotes_list }}"
29       get_url:
30         url: "{{ ma_rnotes_url ~ item ~
31             '/changelogs/CHANGELOG-v' ~ item ~ '.rst' }}"
32         dest: "{{ ma_rnotes_dir }}"
33         owner: "{{ ma_devel_owner }}"
34         group: "{{ ma_devel_group }}"
35         loop: "{{ ma_rnotes_list }}"
36       vars:
37     when: ma_rnotes|bool
38     tags: ma_devel_rnotes
39
40 # EOF
41 ...

```

2.1.5 packages.yml

Synopsis: Configure packages.

Description of the task.

[packages.yml]

```

1 ---
2
3 - name: "packages: Install Ansible packages FreeBSD"
4   pkgng:
5     name: "{{ ma_packages }}"
6     register: result
7     retries: "{{ freebsd_install_retries }}"
8     until: result is succeeded
9     delay: "{{ freebsd_install_delay }}"
10    when:
11      - ansible_os_family == "FreeBSD"
12      - freebsd_install_method|lower == "packages"
13
14 - name: "packages: Install Ansible ports FreeBSD"
15   portinstall:
16     name: "{{ item }}"
17     use_packages: "{{ freebsd_use_packages }}"

```

(continues on next page)

(continued from previous page)

```

18  loop: "{{ ma_packages }}"
19  register: result
20  retries: "{{ freebsd_install_retries }}"
21  until: result is succeeded
22  delay: "{{ freebsd_install_delay }}"
23  when:
24      - ansible_os_family == "FreeBSD"
25      - freebsd_install_method|lower == "ports"
26
27  - name: "packages: Install Ansible packages Linux"
28    package:
29      name: "{{ item }}"
30    loop: "{{ ma_packages }}"
31    register: result
32    retries: "{{ linux_install_retries }}"
33    until: result is succeeded
34    delay: "{{ linux_install_delay }}"
35    when: (ansible_os_family == "Debian") or
36          (ansible_os_family == "RedHat")
37
38  # EOF
39  ...

```

2.1.6 plugins.yml

Synopsis: Configure plugins.

Description of the task.

[plugins.yml]

```

1  ---
2
3  - name: "plugins: Create directories"
4    file:
5      state: directory
6      path: "{{ item }}"
7    loop:
8      - "{{ ma_plugins_path }}"
9      - "{{ ma_src_path }}"
10   tags: ma_plugins_dirs
11
12  - name: "plugins: Download archives"
13    get_url:
14      url: "{{ item.archive_url }}"
15      dest: "{{ ma_src_path }}"
16      checksum: "{{ item.checksum }}"
17    loop: "{{ ma_plugins }}"
18    loop_control:
19      label: "{{ item.archive }}"
20   tags: ma_plugins_download
21
22  - name: "plugins: Extract archives"
23    when: not ansible_check_mode
24    unarchive:
25      src: "{{ ma_src_path }}/{{ item.archive }}"

```

(continues on next page)

(continued from previous page)

```

26     dest: "{{ ma_plugins_path }}"
27     loop: "{{ ma_plugins }}"
28     loop_control:
29         label: "{{ item.archive }}"
30     tags: ma_plugins_extract
31
32 - name: "plugins: Create links"
33   file:
34     state: link
35     src: "{{ ma_plugins_path }}/{{ item.dest }}"
36     dest: "{{ ma_plugins_path }}/{{ item.link }}"
37     loop: "{{ ma_plugins }}"
38     loop_control:
39         label: "{{ item.dest }}"
40   when:
41     - item.link is defined
42     - item.dest is defined
43     - not ansible_check_mode
44   tags: ma_plugins_link
45
46 - name: "plugins: Create lists"
47   block:
48     - name: "plugins: Create list of used ini_keys ma_plugins_inikeys"
49       set_fact:
50         ma_plugins_inikeys: "{{ ma_plugins|json_query('[] .plugins[].ini_key
51 ↪')|list|unique }}"
52     - name: "plugins: Debug ma_plugins_inikeys"
53       when: ma_debug|bool
54       debug:
55         var: ma_plugins_inikeys
56     - name: "plugins: Create list of plugins ma_plugins_list"
57       set_fact:
58         ma_plugins_list: "{{ ma_plugins|json_query('[] .plugins[].{path: path,
59                                                         ini_key: ini_key,
60                                                         enable: enable}'))
61                                                         |list }}"
62     - name: "plugins: Debug ma_plugins_list"
63       when: ma_debug|bool
64       debug:
65         var: ma_plugins_list
66     - name: "plugins: Group plugins by ini_key to ma_plugins_grouped"
67       set_fact:
68         ma_plugins_grouped: "{{ ma_plugins_list|groupby('ini_key') }}"
69     - name: "plugins: Debug ma_plugins_grouped"
70       when: ma_debug|bool
71       debug:
72         var: ma_plugins_grouped
73     - name: "plugins: Create list of ini_key ma_plugins_inikeys_list"
74       set_fact:
75         ma_plugins_inikeys_list: "{{ ma_plugins_sections.keys() }}"
76     - name: "plugins: Debug ma_plugins_inikeys_list"
77       when: ma_debug|bool
78       debug:
79         var: ma_plugins_inikeys_list
80     - name: "plugins: Create lists of paths ma_plugins_paths_list"
81       set_fact:
82         ma_plugins_paths_list: "{{ ma_plugins_paths_list +

```

(continues on next page)

(continued from previous page)

```

82                                     [ {item.0: {'paths': item.1|json_query('[?enable].
↪path')}} } ] }}"
83     loop: "{{ ma_plugins_grouped }}"
84     loop_control:
85         label: "{{ item.0 }}"
86     - name: "plugins: Debug ma_plugins_paths_list"
87       when: ma_debug|bool
88       debug:
89         msg: "{{ item }}"
90       loop: "{{ ma_plugins_paths_list }}"
91       loop_control:
92         label: "{{ item.keys()|list|first }}"
93     tags: ma_plugins_lists
94
95 - name: "plugins: Test used ini_keys"
96   when: item not in ma_plugins_inikeys_list
97   fail:
98     msg: "[ERROR] wrong ini_key: {{ item }}. End of play."
99   loop: "{{ ma_plugins_inikeys }}"
100   tags: ma_plugins_test_inikeys
101
102 # EOF
103 ...

```

2.1.7 vars.yml

Synopsis: Configure vars.

Description of the task.

[vars.yml]

```

1  ---
2
3  - name: "vars: Default vars for
4      {{ ansible_os_family }}
5      {{ ansible_distribution }}
6      {{ ansible_distribution_release }}"
7    include_vars: "{{ item }}"
8    with_first_found:
9      - files:
10        - "{{ ansible_distribution }}-{{ ansible_distribution_release }}.yml"
11        - "{{ ansible_distribution }}.yml"
12        - "{{ ansible_os_family }}.yml"
13        - defaults.yml
14        - default.yml
15        paths: "{{ role_path }}/vars/defaults"
16
17 - name: "vars: Custom vars for
18     {{ ansible_os_family }}
19     {{ ansible_distribution }}
20     {{ ansible_distribution_release }}"
21   include_vars: "{{ item }}"
22   with_first_found:
23     - files:
24       - "{{ ansible_distribution }}-{{ ansible_distribution_release }}.yml"

```

(continues on next page)

(continued from previous page)

```
25     - "{{ ansible_distribution }}.yaml"
26     - "{{ ansible_os_family }}.yaml"
27     - defaults.yaml
28     - default.yaml
29     paths: "{{ role_path }}/vars"
30
31     # EOF
32     ...
```


COPYRIGHT

Redistribution and use in source (RST DocUtils) and ‘compiled’ forms (XML, HTML, PDF, PostScript, RTF and so forth) with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code (RST Docutils) must retain the above copyright notice, this list of conditions and the following disclaimer as the first lines of this file unmodified.

Redistributions in compiled form (transformed to other DTDs, converted to PDF, PostScript, RTF and other formats) must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Important: THIS DOCUMENTATION IS PROVIDED “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE PROVIDER BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS DOCUMENTATION, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

LEGAL NOTICE

All product names, logos, and brands are property of their respective owners.

INDICES AND TABLES

- genindex
- modindex
- search