
ansible-role-freebsd-postinstall Documentation

Release 2.6.1

Vladimir Botka

May 29, 2022

TABLE OF CONTENTS

1	Quick start guide	3
1.1	Requirements on the remote host	3
1.2	Installation	3
1.3	Inventory and playbook	4
1.4	Set hostname and timezone	5
1.5	Install packages	5
1.6	Create users	6
1.7	Update authorized SSH keys	6
1.8	Configure sshd	7
1.9	Configure sudo	8
1.10	Complete role	8
2	User's guide	11
2.1	Introduction	11
2.2	Installation	11
2.3	Playbook	12
2.4	Tags	12
2.5	Debug	14
2.6	Sanity	17
2.7	Tasks	17
2.7.1	mount-img	17
2.7.2	mount-iso	18
2.7.3	passwords	18
2.7.4	users	19
2.8	Development and testing	20
2.9	Examples	20
2.9.1	Example 1: Configure ...	20
2.9.2	Example 2: Configure ...	21
2.10	Variables	22
2.10.1	Default variables	22
2.11	Best practice	22
2.12	Firstboot configuration	23
3	Annotated source code	25
3.1	Tasks	25
3.1.1	main.yml	25
3.1.2	aliases.yml	30
3.1.3	apcupsd.yml	30
3.1.4	authorized-key.yml	31
3.1.5	cloud-providers.yml	33

3.1.6	copy.yml	33
3.1.7	cron.yml	35
3.1.8	debug.yml	37
3.1.9	devfs.yml	39
3.1.10	dhclient.yml	41
3.1.11	fstab.yml	44
3.1.12	gitserver.yml	45
3.1.13	groups.yml	46
3.1.14	groupwrappers.yml	46
3.1.15	hostapd.yml	47
3.1.16	hostname.yml	48
3.1.17	hosts.yml	49
3.1.18	libmap.yml	50
3.1.19	linux.yml	50
3.1.20	loader.yml	52
3.1.21	login.yml	54
3.1.22	make.yml	54
3.1.23	motd.yml	55
3.1.24	mount-img.yml	56
3.1.25	mount-iso.yml	57
3.1.26	mount-vnode.yml	57
3.1.27	nfsd.yml	59
3.1.28	nfs.yml	63
3.1.29	ntpdate.yml	66
3.1.30	ntp.yml	66
3.1.31	packages-install.yml	67
3.1.32	packages.yml	68
3.1.33	passwords.yml	69
3.1.34	periodic.yml	71
3.1.35	procmail.yml	72
3.1.36	resolvconf.yml	74
3.1.37	smartd.yml	75
3.1.38	sanity.yml	76
3.1.39	snmpd.yml	77
3.1.40	sshd.yml	79
3.1.41	ssh.yml	80
3.1.42	sudoers.yml	80
3.1.43	swap.yml	81
3.1.44	sysctl.yml	83
3.1.45	timezone.yml	85
3.1.46	tmpmfs.yml	85
3.1.47	users.yml	86
3.1.48	wpasupplicant.yml	87
3.1.49	digitalocean.yml	89
3.1.50	sync-default.yml	90
3.1.51	sync-optional.yml	91
3.1.52	unarch-default.yml	92
3.1.53	unarch-optional.yml	93
3.1.54	cron-rename-entry.yml	93
3.1.55	mdconfig-attach-disk.yml	94
3.1.56	mdconfig-detach-disk.yml	94
3.1.57	rcconf-item.yml	95
3.2	Handlers	95
3.2.1	main.yml	95

3.2.2	adjkerntz.yml	96
3.2.3	apcupsd.yml	97
3.2.4	devfs.yml	97
3.2.5	dhclient.yml	97
3.2.6	fstab.yml	98
3.2.7	hostapd.yml	98
3.2.8	hostname.yml	99
3.2.9	login.yml	99
3.2.10	messages.yml	99
3.2.11	newaliases.yml	99
3.2.12	nfsd.yml	100
3.2.13	nfs.yml	101
3.2.14	ntpdate.yml	101
3.2.15	ntpd.yml	101
3.2.16	qemu.yml	102
3.2.17	resolvconf.yml	102
3.2.18	smartd.yml	102
3.2.19	snmpd.yml	103
3.2.20	sshd.yml	103
3.2.21	swap.yml	104
3.2.22	sysctl.yml	105
3.2.23	wpasupplicant.yml	105
3.3	Templates	105
3.3.1	apcupsd.conf.j2	105
3.3.2	exports.j2	106
3.3.3	groupadd.j2	106
3.3.4	groupdel.j2	107
3.3.5	groupmod.j2	107
3.3.6	hostapd.conf.j2	107
3.3.7	hosts.j2	108
3.3.8	hosts-aliases-list.j2	108
3.3.9	loader.conf.j2	109
3.3.10	login.conf.j2	109
3.3.11	login_conf.j2	110
3.3.12	make.conf.j2	110
3.3.13	motd.j2	110
3.3.14	ntp.conf.j2	111
3.3.15	procmailrc-includerc-capability.j2	112
3.3.16	procmailrc-includerc-duplicates.j2	112
3.3.17	procmailrc-includerc-get-certificate.j2	113
3.3.18	procmailrc-includerc-get-fingerprint.j2	113
3.3.19	procmailrc-includerc-get-fingerprints.j2	113
3.3.20	procmailrc-includerc-get-message.j2	114
3.3.21	ssh_config.j2	114
3.3.22	wpa_supplicant.conf.j2	115
4	Copyright	117
5	Legal Notice	119
6	Indices and tables	121
	Index	123

This [role](#) and the documentation is work in progress. If the documentation of a task is missing it's necessary to review the [source code](#) to learn how to use it. If a functionality is missing consider role [config_light](#). See various [examples](#). If *config_light* is not able to do what you want create new tasks.

Feel free to [share your feedback and report issues](#).

Contributions are welcome.

QUICK START GUIDE

For the users who want to try the role quickly this guide provides an example of how to:

- Configure Ansible and install the role
- Create inventory and playbook

Then, on the controller, run the playbook and configure the remote host:

- Set hostname and timezone
- Install packages
- Create users and update authorized SSH keys
- Configure sshd and sudo.

1.1 Requirements on the remote host

- Password-less ssh by *admin* (or any other user by your choice)
- Login shell for *admin* is `/bin/sh`
- User *admin* is granted unlimited sudo
- Python and Perl are installed.

See also:

- [Ansible and BSD](#)
- [Installing Ansible on FreeBSD](#)

1.2 Installation

Configure Ansible. Fit the paths to the inventory (3) and to the roles (4) to your needs. Let's disable *retry_files* (5). Don't display *ok* and *skipped* hosts (6-8). Pipelining (12) should speedup the execution of the playbook

```
1 shell> cat ansible.cfg
2 [defaults]
3 inventory = $PWD/hosts
4 roles_path = $HOME/.ansible/roles
5 retry_files_enabled = false
6 stdout_callback = default
7 display_ok_hosts=false
```

(continues on next page)

(continued from previous page)

```
8 display_skipped_hosts=false
9 log_path = /var/log/ansible.log
10
11 [ssh_connection]
12 pipelining = true
```

Install the role `vbotka.freebsd_postinstall`

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

Install the library `vbotka.ansible_lib` and the collections

```
shell> ansible-galaxy role install vbotka.ansible_lib
shell> ansible-galaxy collection install ansible.posix
shell> ansible-galaxy collection install community.general
```

Review the role

```
1 shell> ls ~/.ansible/roles/vbotka.freebsd_postinstall
2 contrib defaults docs handlers LICENSE meta README.md
3 tasks templates tests vars
```

1.3 Inventory and playbook

Create the inventory hosts

```
1 shell> cat hosts
2 test_14.example.com ansible_host=10.1.0.64
3
4 [example_com]
5 test_14.example.com
6
7 [example_com:vars]
8 ansible_python_interpreter=/usr/local/bin/python3.8
9 ansible_perl_interpreter=/usr/local/bin/perl
```

Create the playbook `playbook.yml` for single host `test_14.example.com`

```
1 shell> cat playbook.yml
2 - hosts: test_14.example.com
3   gather_facts: true
4   connection: ssh
5   remote_user: admin
6   become: true
7   become_user: root
8   become_method: sudo
9   roles:
10    - vbotka.freebsd_postinstall
```

Test syntax of the playbook

```

1 shell> ansible-playbook playbook.yml --syntax-check
2
3 playbook: playbook.yml

```

1.4 Set hostname and timezone

Create `host_vars` with customized variables. Enable `hostname` (2) and `timezone` (5). Disable debug output (7) and backup of changed files (8)

```

1 shell> cat host_vars/test_14.example.com/fp-common.yml
2 fp_hostname: true
3 rc_conf_hostname: test_14.example.com
4 fp_domain: example.com
5 fp_timezone: true
6 fp_zoneinfo: UTC
7 fp_debug: false
8 fp_backup_conf: false

```

Set hostname and timezone

```

1 shell> ansible-playbook playbook.yml -t fp_hostname,fp_timezone
2
3 PLAY [test_14.example.com]
4 ↪*****
5 TASK [vbotka.freebsd_postinstall : hostname: Configure hostname in /etc/rc.conf]
6 ↪*****
7 changed: [test_14.example.com] => (item={'key': 'hostname', 'value': 'test_14.example.com
8 ↪'})
9
10 RUNNING HANDLER [vbotka.freebsd_postinstall : set hostname]
11 ↪*****
12 changed: [test_14.example.com]
13
14 PLAY RECAP
15 ↪*****
16 test_14.example.com: ok=4 changed=2 unreachable=0 failed=0 skipped=22 rescued=0 ignored=0

```

UTC is the default timezone. Therefore, no changes were reported.

1.5 Install packages

Enable the installation of packages (2) and pick the `lists` to be installed (3). See available `lists` in `defaults/main/pkgdict_*.yml`

```

1 shell> cat host_vars/test_14.example.com/fp-packages.yml
2 fp_install: true
3 fp_packages:
4   - {list: minimal, enabled: true}

```

Install packages

```
1 shell> ansible-playbook playbook.yml -t fp_packages
2 <TODO>
```

1.6 Create users

Enable the management of users (2) and create the configuration data (3). Delete default user (7)

```
1 shell> cat host_vars/test_14.example.com/fp-users.yml
2 fp_users: true
3 fp_users_conf:
4   - {name: admin, shell: /bin/sh, groups: [wheel]}
5   - {name: devel, shell: /usr/local/bin/bash, groups: [wheel]}
6   - {name: responder, shell: /usr/sbin/nologin}
7   - {name: freebsd, state: absent}
```

Manage users

```
1 shell> ansible-playbook playbook.yml -t fp_users
2
3 PLAY [test_14.example.com]
4   ↳ *****
5 TASK [vbotka.freebsd_postinstall : users: Manage user accounts]
6   ↳ *****
7 changed: [test_14.example.com] => (item=admin)
8 changed: [test_14.example.com] => (item=devel)
9 changed: [test_14.example.com] => (item=responder)
10
11 PLAY RECAP
12   ↳ *****
13 test_14.example.com: ok=2 changed=1 unreachable=0 failed=0 skipped=22 rescued=0 ignored=0
```

1.7 Update authorized SSH keys

Enable the installation of authorized keys (2). Remove other keys if already configured (3). Install the keys in batches (4). Create the list of the public keys to be installed (5)

```
1 shell> cat host_vars/test_14.example.com/fp-authorized-key.yml
2 fp_authorized_key: true
3 fp_authorized_key_install_exclusive: true
4 fp_authorized_key_install_individually: false
5 fp_authorized_key_conf:
6   - user: admin
7     key: "{{ lookup('file', '~/ansible/ssh-pub-keys/devel-srv1/id_rsa.pub') }}"
8   - user: admin
9     key: "{{ lookup('file', '~/ansible/ssh-pub-keys/devel-srv2/id_rsa.pub') }}"
```

Update authorized keys

```
1 shell> ansible-playbook playbook.yml -t fp_authorized_key
2 <TODO>
```

1.8 Configure sshd

Enable the configuration of *sshd* (2), enable *sshd* service, and create the configuration data (10)

```
1 shell> cat host_vars/test_14.example.com/fp-sshd.yml
2 fp_sshd: true
3 fp_sshd_enable: true
4 fp_sshd_passwordauthentication: "no"
5 fp_sshd_challengeresponseauthentication: "no"
6 fp_sshd_permitrootlogin: "no"
7 fp_sshd_permittunnel: "no"
8 fp_sshd_allowtcpforwarding: "yes"
9 fp_sshd_x11forwarding: "no"
10 fp_sshd_conf:
11   - {key: PasswordAuthentication, value: "{{ fp_sshd_passwordauthentication }}" }
12   - {key: ChallengeResponseAuthentication, value: "{{ fp_sshd_
13     ↳ challengeresponseauthentication }}" }
14   - {key: PermitRootLogin, value: "{{ fp_sshd_permitrootlogin }}" }
15   - {key: PermitTunnel, value: "{{ fp_sshd_permittunnel }}" }
16   - {key: AllowTcpForwarding, value: "{{ fp_sshd_allowtcpforwarding }}" }
17   - {key: X11Forwarding, value: "{{ fp_sshd_x11forwarding }}" }
18   - {key: UseBlacklist, value: "yes" }
```

Configure *sshd*

```
1 shell> ansible-playbook playbook.yml -t fp_sshd
2
3 PLAY [test_14.example.com]_
4   ↳ *****
5 TASK [vbotka.freebsd_postinstall : sshd: Configure /etc/ssh/sshd_config]_
6   ↳ *****
7 changed: [test_14.example.com] => (item={'key': 'PasswordAuthentication', 'value': 'no'})
8 changed: [test_14.example.com] => (item={'key': 'ChallengeResponseAuthentication', 'value
9   ↳ ': 'no'})
10 changed: [test_14.example.com] => (item={'key': 'PermitRootLogin', 'value': 'no'})
11 changed: [test_14.example.com] => (item={'key': 'PermitTunnel', 'value': 'no'})
12 changed: [test_14.example.com] => (item={'key': 'AllowTcpForwarding', 'value': 'yes'})
13 changed: [test_14.example.com] => (item={'key': 'X11Forwarding', 'value': 'no'})
14 changed: [test_14.example.com] => (item={'key': 'UseBlacklist', 'value': 'yes'})
15
16 RUNNING HANDLER [vbotka.freebsd_postinstall : reload sshd]_
17   ↳ *****
18 changed: [test_14.example.com]
19
20 PLAY RECAP_
21   ↳ *****
22 test_14.example.com: ok=4 changed=2 unreachable=0 failed=0 skipped=23 rescued=0 ignored=0
```

1.9 Configure sudo

Enable the configuration of *sudoers* (2) and create the configuration data (3)

```

1 shell> cat host_vars/test_14.example.com/fp-sudoers.yml
2 fp_sudoers: true
3 fp_sudoers_conf:
4   - {key: admin, value: "ALL=(ALL) NOPASSWD: ALL"}
5   - {key: devel, value: "ALL=(ALL) NOPASSWD: ALL"}

```

Configure *sudoers*

```

1 shell> ansible-playbook playbook.yml -t fp_sudoers
2
3 PLAY [test_14.example.com]
4 ↪ *****
5 TASK [vbotka.freebsd_postinstall : sudoers: Configure /usr/local/etc/sudoers]
6 ↪ *****
7 changed: [test_14.example.com] => (item={'key': 'devel', 'value': 'ALL=(ALL) NOPASSWD:
8 ↪ ALL'})
9
10 PLAY RECAP
11 ↪ *****
12 test_14.example.com: ok=2 changed=1 unreachable=0 failed=0 skipped=22 rescued=0 ignored=0

```

The user *admin* has already been enabled. Otherwise the Ansible escalation become: `true` wouldn't work. Therefore, only the user *devel* reported changes.

1.10 Complete role

To show the functionality of the role, the tasks in the previous examples (*fp_hostname*, *fp_timezone*, *fp_packages*, *fp_users*, *fp_authorized_key*, *fp_sshd*, *fp_sudoers*) were run separately. You can create the data and run the tasks all in one play

```

1 shell> tree .
2
3 ├── ansible.cfg
4 ├── playbook.yml
5 ├── hosts
6 └── host_vars
7     └── test_14.example.com
8         ├── fp-authorized-key.yml
9         ├── fp-common.yml
10        ├── fp-packages.yml
11        ├── fp-sshd.yml
12        ├── fp-sudoers.yml
13        └── fp-users.yml
14
15 2 directories, 9 files

```

Run all enabled tasks in the playbook again. Optionally, disable installation to speedup the execution

```

1 shell> ansible-playbook playbook.yml -e fp_install=false
2
3 PLAY [test_14.example.com]
4 ↳*****
5 PLAY RECAP
6 ↳*****
test_14.example.com: ok=10 changed=0 unreachable=0 failed=0 skipped=172 rescued=0
↳ignored=0

```

Warning: The host has not been secured by this playbook and should be used for testing only.

2.1 Introduction

The role is a collection of independent sets of tasks, grouped in separate files, that configure particular functionality of the system (cron, hostname, sshd, ...). By default all tasks are disabled. See the control variables of the form `fp_<tasks_group>` in `tasks/main.yml`. There are two more categories of control variables used by some of the tasks. `fp_<tasks_group>_install` and `fp_<task_group>_enable`. The first one can be used to selectively install predefined groups of packages needed by the tasks. See `tasks/packages.yml` and `defaults/main/pkgdict_*.yml`. The control variables `fp_<tasks_group>_enable` can be used to enable or disable the particular services. Configure and enable the tasks and services that you need. In addition to this, some groups of tasks start with the debug output which can be turned on by enabling `fp_<tasks_group>_debug`. See the Debug section below. To speedup the execution of a playbook it's possible to select particular groups of tasks only. See the Tags section on how to use them. Some tasks are described in detail in the section Tasks. If the description of a task is missing it's necessary to review the source code on how to use it. To learn details, it's always recommended to review the code and notes in the section Annotated source code. See the section Best practice on how to check syntax, preview potential changes, and debug tasks.

- Ansible role: [freebsd_postinstall](#)
- Supported systems: [FreeBSD](#)
- Requirements: [ansible_lib](#)

2.2 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 role info vbotka.freebsd_postinstall
```

and install it

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

Install the requirements

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

Install the collections [community.general](#) and [ansible.posix](#)

```
shell> ansible-galaxy collection install ansible.posix
shell> ansible-galaxy collection install community.general
```

See also:

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

Note: This role configures FreeBSD on the remote host. To install and configure Ansible on the controller running FreeBSD see:

- [Ansible and BSD - docs.ansible.com](#)
 - [Installing Ansible on FreeBSD - docs.ansible.com](#)
 - [Install and configure Ansible - Ansible role at galaxy.ansible.com](#)
-

2.3 Playbook

Create simple playbook that calls this role (10) at a single host `test_14.example.com` (2)

```
1 shell> cat playbook.yml
2 - hosts: test_14.example.com
3   gather_facts: true
4   connection: ssh
5   remote_user: admin
6   become: true
7   become_user: root
8   become_method: sudo
9   roles:
10  - vbotka.freebsd_postinstall
```

Note: Enable `gather_facts: true` (3) to collect facts.

See also:

- For details see [Connection Plugins](#) (4-5)
- See also [Understanding Privilege Escalation](#) (6-8)

2.4 Tags

The tags provide the user with a very useful tool to run selected groups of tasks. Use the command below to display available tags

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

playbook: playbook.yml
```

(continues on next page)

(continued from previous page)

```

play #1 (test_14.example.com): test_14.example.com TAGS: []

TASK TAGS: [always, fp_aliases, fp_apcupsd, fp_apcupsd_conf,
fp_apcupsd_rcconf, fp_authorized_key, fp_authorized_key_debug,
fp_cloud_providers, fp_copy, fp_copy_debug,
fp_copy_sync_default, fp_copy_sync_optional,
fp_copy_unarch_default, fp_copy_unarch_optional, fp_cron,
fp_cron_debug, fp_cron_rename_entries,
fp_cron_rename_entries_system, fp_cron_tab, fp_cron_tab_system,
fp_cron_var, fp_cron_var_system, fp_debug, fp_devfs,
fp_devfs_conf, fp_devfs_debug, fp_devfs_rcconf, fp_devfs_rcvar,
fp_devfs_rules, fp_do, fp_do_debug, fp_do_default_user_remove,
fp_do_hostname_rcconf, fp_do_rcconf, fp_fstab, fp_gitserver,
fp_gitserver_dir, fp_gitserver_group, fp_gitserver_owner,
fp_groups, fp_groups_conf, fp_groups_debug, fp_groupwrappers,
fp_groupwrappers_dir, fp_groupwrappers_groupadd,
fp_groupwrappers_groupdel, fp_groupwrappers_groupmod,
fp_hostapd, fp_hostapd_conf, fp_hostapd_rcconf, fp_hostname,
fp_hosts, fp_hosts_conf, fp_hosts_debug, fp_hosts_sanity,
fp_libmap, fp_linux, fp_linux_debug, fp_linux_fstab,
fp_linux_lib_copy, fp_linux_lib_dir, fp_linux_lib_link,
fp_linux_rcconf, fp_loader, fp_loader_debug, fp_login, fp_make,
fp_make_conf, fp_make_debug, fp_motd, fp_motd_create,
fp_motd_rcconf, fp_mount_iso, fp_mount_iso_debug, fp_nfs,
fp_nfs_lockd_rcconf, fp_nfs_rcconf, fp_nfs_statd_rcconf,
fp_nfsd, fp_ntp, fp_ntp_conf, fp_ntp_rcconf, fp_ntpdate,
fp_packages, fp_packages_debug, fp_packages_links, fp_passwords,
fp_passwords_debug, fp_passwords_passwordstore, fp_periodic,
fp_periodic_conf_local, fp_periodic_debug, fp_procmail,
fp_procmail_dir, fp_procmail_forward, fp_procmail_forwarders,
fp_procmail_includerc, fp_procmail_procmailrc,
fp_resolv_conf_absent, fp_resolvconf, fp_resolvconf_absent,
fp_resolvconf_conf, fp_smartd, fp_smartd_conf, fp_smartd_rcconf,
fp_snmpd, fp_snmpd_conf, fp_snmpd_conf_copy, fp_snmpd_rcconf,
fp_ssh, fp_ssh_config, fp_ssh_debug, fp_sshd, fp_sshd_config,
fp_sshd_debug, fp_sshd_rcconf, fp_sudoers, fp_swap,
fp_swap_debug, fp_sysctl, fp_timezone, fp_timezone_debug,
fp_tmpmfs, fp_tmpmfs_debug, fp_tmpmfs_rcconf, fp_users,
fp_users_debug, fp_wpasupplicant, fp_wpasupplicant_conf,
fp_wpasupplicant_conf_file_link, fp_wpasupplicant_conf_rc,
fp_wpasupplicant_debug, fp_wpasupplicant_rcconf]

```

For example,

- enable debug and display the list of the control variables and their values

```
shell> ansible-playbook playbook.yml -t fp_debug -e fp_debug=true
```

- See what packages will be installed

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

- Install the packages only

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

2.5 Debug

To display some facts (11-16) and values of the control variables, run select the task `fp_debug` and enable the debug output. By default all control variables are disabled. The example below shows values used in the examples of the Quick start guide

```

1 shell> ansible-playbook playbook.yml -t fp_debug -e fp_debug=true
2
3 PLAY [test_14.example.com]_
4 ↪ *****
5 TASK [Gathering Facts]_
6 ↪ *****
7 ok: [test_14.example.com]
8
9 TASK [vbotka.freebsd_postinstall : FreeBSD postinstall Debug]_
10 ↪ *****
11 ok: [test_14.example.com] => {
12   msg: [
13     - ansible_architecture [amd64]
14     - ansible_os_family [FreeBSD]
15     - ansible_distribution [FreeBSD]
16     - ansible_distribution_major_version [12]
17     - ansible_distribution_version [12.1]
18     - ansible_distribution_release [12.1-RELEASE]
19     - ''
20     - rc_conf_hostname [test_14.example.com]
21     - fp_domain [example.com]
22     - aws [False]
23     - digitalocean [False]
24     - gcp [False]
25     - ''
26     - freebsd_install_method [packages]
27     - freebsd_use_packages [True]
28     - freebsd_install_retries [3]
29     - freebsd_install_delay [5]
30     - ''
31     - fp_install [False]
32     - fp_backup [False]
33     - fp_sanity [True] fp_sanity_tags_fatal [True]
34     - ''
35     - fp_aliases [True]
36     - fp_apcupsd [False] fp_apcupsd_install [False] fp_apcupsd_enabled [False]
37     - fp_authorized_key [True]
38     - fp_copy [True] fp_unarch_default [False] fp_unarch_optional [False]
39     - fp_cron [True]
40     - fp_devfs [False] fp_devfs_enabled [False]
41     - fp_fstab [True]
42     - fp_gitserver [False] fp_gitserver_install [False]

```

(continues on next page)

(continued from previous page)

```

41 - fp_groups [True]
42 - fp_groupwrappers_install [True]
43 - fp_hostapd [False] fp_hostapd_install [False] fp_hostapd_enabled [False]
44 - fp_hostname [True]
45 - fp_linux [True] fp_linux_install [False] fp_linux_enable [True]
46 - fp_loader [True]
47 - fp_login [True]
48 - fp_make [True]
49 - fp_motd [True]
50 - fp_mount_img [False]
51 - fp_mount_iso [False]
52 - fp_nfs [True] fp_nfs_enable [True]
53 - fp_nfsd [True] fp_nfsd_enable [False]
54 - fp_ntp [True] fp_ntp_enable [True]
55 - fp_ntpdate [True]
56 - fp_passwords [True]
57 - fp_periodic [True]
58 - fp_procmail [True] fp_procmail_install [True]
59 - fp_resolvconf [False]
60 - fp_smartd [True] fp_smartd_install [True] fp_smartd_enabled [True]
61 - fp_snmpd [False] fp_snmpd_install [False] fp_snmpd_enabled [False]
62 - fp_ssh [True]
63 - fp_sshd [True] fp_sshd_enable [True]
64 - fp_sudoers [True]
65 - fp_swap [False] fp_swap_enable [False]
66 - fp_sysctl [True]
67 - fp_timezone [True]
68 - fp_tmpmfs [False]
69 - fp_users [True]
70 - fp_wpasuppliant [True] fp_wpasuppliant_install [True]
71 - ''
72 - fp_mountd_enable [True]
73 - fp_rpcbind_enable [True]
74 - fp_rpc_lockd_enable [True]
75 - fp_rpc_statd_enable [True]
76 ]
77 }
78
79 PLAY RECAP
80 ↪ *****
test_14.example.com: ok=2 changed=0 unreachable=0 failed=0 skipped=21 rescued=0 ignored=0

```

Note: If you use the configuration from the Quick start guide enable *ok* hosts either by the environment variable `ANSIBLE_DISPLAY_OK_HOSTS=True` or in the configuration file `display_ok_hosts=true`.

Some tasks will display additional information when the variable `fp_*_debug` is enabled, e.g. configuration of `sshd` below. Inspect the files in the tasks directory whether *debug* is available or not

```

1 shell> ansible-playbook playbook.yml -t fp_sshd_debug -e fp_sshd_debug=true
2
3 PLAY [test_14.example.com]
↪ ***** (continues on next page)

```

(continued from previous page)

```

4
5 TASK [Gathering Facts] 
  ↪*****
6 ok: [test_14.example.com]
7
8 TASK [vbotka.freebsd_postinstall : sshd: Debug] 
  ↪*****
9 ok: [test_14.example.com] => {
10     "msg": [
11         "fp_sshd_bin [/usr/sbin/sshd]",
12         "fp_sshd_conf_file [/etc/ssh/sshd_config]",
13         "fp_sshd_conf",
14         "- {key: PasswordAuthentication, value: 'no'}",
15         "- {key: ChallengeResponseAuthentication, value: 'no'}",
16         "- {key: PermitRootLogin, value: 'no'}",
17         "- {key: PermitTunnel, value: 'no'}",
18         "- {key: AllowTcpForwarding, value: 'yes'}",
19         "- {key: X11Forwarding, value: 'no'}",
20         "- {key: UseBlacklist, value: 'yes'}"
21     ]
22 }
23
24 PLAY RECAP 
  ↪*****
25 test_14.example.com: ok=2 changed=0 unreachable=0 failed=0 skipped=21 rescued=0 ignored=0

```

Note:

- The debug output of this role is optimized for the *default* or *yaml* callback plugin. Set the plugin in the environment, e.g. `shell> export ANSIBLE_STDOUT_CALLBACK=default` or in the configuration file `stdout_callback = default`.
- See the details about the *yaml* callback plugin `shell> ansible-doc -t callback yaml`
- See the list of other callback plugins `shell> ansible-doc -t callback -l`

See also:

- [Playbook Debugger](#)
- [Debugging modules](#)
- [Python Debugging With Pdb](#)

2.6 Sanity

By default the checking of sanity is turned on

```
fp_sanity: true
fp_sanity_tags_fatal: true
```

<TODO>

2.7 Tasks

The description of the tasks is not complete. The `role` and the documentation is work in progress. Feel free to [share your feedback](#) and report issues.

Contributions are welcome.

See also:

- Source code `main.yml`

Table of Contents

2.7.1 mount-img

Mount images.

Example

Enable the mounting of images (2) and create the list of mount points (3). For example

```
1 shell> cat host_vars/build.example.com/fp-mount-img.yml
2 fp_mount_img: true
3 fp_mount_img_entries:
4   - img: /export/img/FreeBSD-12.2-RELEASE-arm-armv7-CUBOX-HUMMINGBOARD.img
5     mount: /usr/local/poudriere/jails/12arm7
6     fstype: ufs
7     partition: s2a
```

Dry-run the tasks, display the variables, and display the changes

```
shell> ansible-playbook playbook.yml -t fp_mount_img -e fp_mount_img_debug=true -CD
```

If all seems to be right configure the vnode and mount the image

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

Log to the remote host, list memory disks (2) and related mount points (5)

```
1 shell> mdconfig -l -v
2 md0 vnode 3072M /export/img/FreeBSD-12.2-RELEASE-arm-armv7-CUBOX-HUMMINGBOARD.img
3
4 shell> df | grep arm
5 /dev/md0s2a 5971064 4056912 1436472 74% /zroot/poudriere/jails/12arm7
```

See also:

- Annotated Source code [mount-img.yml](#)

2.7.2 mount-iso

Mount images.

Example

Enable the mounting of ISO images (2) and create the list of mount points (3). For example

```
1 shell> cat host_vars/build.example.com/fp-mount-iso.yml
2 fp_mount_iso: true
3 fp_mount_iso_entries:
4   - iso: /export/img/FreeBSD/12.2/FreeBSD-12.2-RELEASE-amd64-disc1.iso
5     mount: /export/distro/FreeBSD-12.1-RELEASE-amd64-disc1.iso
```

Dry-run the tasks, display the variables, and display the changes

```
shell> ansible-playbook playbook.yml -t fp_mount_iso -e fp_mount_iso_debug=true -CD
```

If all seems to be right create the vnode and mount the ISO image

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

Log to the remote host, list memory disks (2) and related mount points (5)

```
1 shell> mdconfig -l -v
2 md0 vnode 868M /export/images/FreeBSD/12.2/FreeBSD-12.2-RELEASE-amd64-disc1.iso
3
4 shell> df | grep iso
5 /dev/md0 888380 888380 0 100% /export/distro/FreeBSD-12.2-RELEASE-amd64-disc1.iso
```

See also:

- Annotated Source code [mount-iso.yml](#)

2.7.3 passwords

Manage users' passwords. At the moment only passwordstore *Unix password manager* is implemented.

Passwordstore

Create passwords in passwordstore and enable access for the user who is running the playbook. For example

```
shell> pass build.example.com/
build.example.com
├─ admin
└─ root
```

Configure the variables. Enable the management of the users' passwords (3), select passwordstore (4), and configure the variable that keeps the name of the host used in passwordstore (5), `ansible_nodename` in this case. Next options might be `inventory_hostname` (default), `ansible_hostname`, or any other custom variable. For example


```
1 shell> cat host_vars/build.example.com/fp-passwords.yml
2 ---
3 fp_passwords: true
4 fp_passwordstore: true
5 fp_passwordstore_hostname_var: ansible_nodename
6 ...
```

Dry-run the tasks, display the variables, and display the changes

```
shell> ansible-playbook playbook.yml -t fp_passwords -e fp_passwords_debug=true -CD
```

If all seems to be right manage the users' passwords

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

See also:

- Annotated Source code [passwords.yml](#)
- Project website passwordstore.org

2.7.4 users

Manage users.

Example

Enable management of users (3) and create list of the users (4). For example

```
1 shell> cat host_vars/build.example.com/fp-users.yml
2 ---
3 fp_users: true
4 fp_users_conf:
5   - name: admin
6     shell: /bin/sh
7     groups: [wheel]
8   - name: responder
9     shell: /usr/sbin/nologin
10  - name: freebsd
11    state: absent
12 ...
```

Dry-run the tasks, display the variables, and display the changes

```
shell> ansible-playbook playbook.yml -t fp_users -e fp_users_debug=true -CD
```

If all seems to be right configure the users

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

See also:

- Annotated Source code [users.yml](#)

2.8 Development and testing

In order to deliver an Ansible project it's necessary to test the code and configuration. The tags provide the administrators with a tool to test groups of tasks. For example, to test a group of tasks at single remote host *test_01*, create the playbook

```
shell> cat playbook.yml
- hosts: test_01
  roles:
    - vbotka.freebsd_postinstall
```

Customize configuration in `host_vars/test_01/fp-*.yml` and check the syntax

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

Then dry-run the selected group of tasks and see what will be changed. Replace `<tag>` with valid tag or with a comma-separated list of tags

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

<TBC>

2.9 Examples

<TBD>

Table of Contents

2.9.1 Example 1: Configure ...

Create a playbook

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

Create `host_vars/test_01/XY-task1.yml`

```
shell> cat host_vars/test_01/XY-task1.yml
XY_task1: true
...
```

Note:

- When ...

- See ...

Configure ...

```
shell> ansible-playbook playbook.yml -t XY_task1

TASK [vbotka.freebsd_postinstall : task1: Configure ...] **
ok: [test_01] => (...)
```

The command is idempotent

```
shell> ansible-playbook playbook.yml -t XY_task1
...
PLAY RECAP *****
test_01: ok=6 changed=0 unreachable=0 failed=0 skipped=4 rescued=0 ignored=0
```

2.9.2 Example 2: Configure ...

Create a playbook

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

Create *host_vars/test_01/XY-task1.yml*

```
shell> cat host_vars/test_01/XY-task1.yml
XY_task1: true
...
```

Note:

- When ...
- See ...

Configure ...

```
shell> ansible-playbook playbook.yml -t XY_task1

TASK [vbotka.freebsd_postinstall : task1: Configure ...] **
ok: [test_01] => (...)
```

The command is idempotent

```
shell> ansible-playbook playbook.yml -t XY_task1
...
PLAY RECAP *****
test_01: ok=6 changed=0 unreachable=0 failed=0 skipped=4 rescued=0 ignored=0
```

2.10 Variables

See also:

- [Ansible variable precedence: Where should I put a variable?](#)

2.10.1 Default variables

[defaults/main]

The common variables are stored in the files in the directory `defaults/main`. These variables can be customized in the file `vars/main.yml`. The file `vars/main.yml` will be preserved by the update of the role.

Warning:

- Don't make any changes to the files in the directory `defaults/main`. The changes might be overwritten when the role is updated. Customize the default values in the file `vars/main.yml`.

See also:

- The examples of the customization in the directory `vars/samples`

2.11 Best practice

Test syntax

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

See what variables will be included

```
shell> ansible-playbook playbook.yml -t fp_debug -e fp_debug=true
```

Dry-run, display differences and display variables

```
shell> ansible-playbook playbook.yml -e fp_debug=true --check --diff
```

Install packages. Run this task separately if the installation takes too long. Then disable it.

```
shell> ansible-playbook playbook.yml -t fp_packages -e fp_install=true
```

Run the playbook

```
shell> ansible-playbook playbook.yml -e fp_install=false
```

Test the idempotency. The role and the configuration data shall be idempotent. There should be no changes reported by *ansible-playbook* when the playbook is run repeatedly. Disable the debug output, and the installation of packages or ports to speedup the playbook.

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

2.12 Firstboot configuration

Configure hostname, users, sudoers, network and reboot

```
shell> ansible-playbook playbook.yml -t fp_hostname, ...
```

<TBD>

ANNOTATED SOURCE CODE

3.1 Tasks

3.1.1 main.yml

Synopsis: Main task.

Import tasks if enabled.

[tasks/main.yml]

```
1 ---
2 # tasks for freebsd_postinstall
3
4 - name: Import debug.yml
5   ansible.builtin.import_tasks: debug.yml
6   when: fp_debug|bool
7   tags: fp_debug
8
9 - name: Import sanity.yml
10  ansible.builtin.import_tasks: sanity.yml
11  when: fp_sanity|bool
12  tags: always
13
14 - name: Import cloud-providers.yml
15  ansible.builtin.import_tasks: cloud-providers.yml
16  tags: fp_cloud_providers
17
18 - name: Import loader.yml
19  ansible.builtin.import_tasks: loader.yml
20  when: fp_loader|bool
21  tags: fp_loader
22
23 - name: Import sysctl.yml
24  ansible.builtin.import_tasks: sysctl.yml
25  when: fp_sysctl|bool
26  tags: fp_sysctl
27
28 - name: Import tmpmfs.yml
29  ansible.builtin.import_tasks: tmpmfs.yml
30  when: fp_tmpmfs|bool
```

(continues on next page)

(continued from previous page)

```
31 tags: fp_tmpmfs
32
33 - name: Import devfs.yml
34   ansible.builtin.import_tasks: devfs.yml
35   when: fp_devfs|bool
36   tags: fp_devfs
37
38 - name: Import dhclient.yml
39   ansible.builtin.import_tasks: dhclient.yml
40   when: fp_dhclient|bool
41   tags: fp_dhclient
42
43 - name: Import make.yml
44   ansible.builtin.import_tasks: make.yml
45   when: fp_make|bool
46   tags: fp_make
47
48 - name: Import swap.yml
49   ansible.builtin.import_tasks: swap.yml
50   when: fp_swap|bool
51   tags: fp_swap
52
53 - name: Import flush_handlers
54   ansible.builtin.meta: flush_handlers
55
56 - name: Import fstab.yml
57   ansible.builtin.import_tasks: fstab.yml
58   when: fp_fstab|bool
59   tags: fp_fstab
60
61 - name: Import mount-iso.yml
62   ansible.builtin.import_tasks: mount-iso.yml
63   when: fp_mount_iso|bool
64   tags: fp_mount_iso
65
66 - name: Import mount-img.yml
67   ansible.builtin.import_tasks: mount-img.yml
68   when: fp_mount_img|bool
69   tags: fp_mount_img
70
71 - name: Import packages.yml
72   ansible.builtin.import_tasks: packages.yml
73   when: fp_install|bool
74   tags: fp_packages
75
76 - name: Import copy.yml
77   ansible.builtin.import_tasks: copy.yml
78   when: fp_copy|bool
79   tags: fp_copy
80
81 - name: Import libmap.yml
82   ansible.builtin.import_tasks: libmap.yml
```

(continues on next page)

(continued from previous page)

```
83  when: fp_libmap|bool
84  tags: fp_libmap
85
86  - name: Import groups.yml
87    ansible.builtin.import_tasks: groups.yml
88    when: fp_groups|bool
89    tags: fp_groups
90
91  - name: Import users.yml
92    ansible.builtin.import_tasks: users.yml
93    when: fp_users|bool
94    tags: fp_users
95
96  - name: Import passwords.yml
97    ansible.builtin.import_tasks: passwords.yml
98    when: fp_passwords|bool
99    tags: fp_passwords
100
101  - name: Import groupwrappers.yml
102    ansible.builtin.import_tasks: groupwrappers.yml
103    when: fp_groupwrappers|bool
104    tags: fp_groupwrappers
105
106  - name: Import hostname.yml
107    ansible.builtin.import_tasks: hostname.yml
108    when: fp_hostname|bool
109    tags: fp_hostname
110
111  - name: Import hosts.yml
112    ansible.builtin.import_tasks: hosts.yml
113    when: fp_hosts|bool
114    tags: fp_hosts
115
116  - name: Import ssh.yml
117    ansible.builtin.import_tasks: ssh.yml
118    when: fp_ssh|bool
119    tags: fp_ssh
120
121  - name: Import sshd.yml
122    ansible.builtin.import_tasks: sshd.yml
123    when: fp_sshd|bool
124    tags: fp_sshd
125
126  - name: Import authorized-key.yml
127    ansible.builtin.import_tasks: authorized-key.yml
128    when: fp_authorized_key|bool
129    tags: fp_authorized_key
130
131  - name: Import timezone.yml
132    ansible.builtin.import_tasks: timezone.yml
133    when: fp_timezone|bool
134    tags: fp_timezone
```

(continues on next page)

(continued from previous page)

```
135
136 - name: Import ntp.yml
137   ansible.builtin.import_tasks: ntp.yml
138   when: fp_ntp|bool
139   tags: fp_ntp
140
141 - name: Import ntpdate.yml
142   ansible.builtin.import_tasks: ntpdate.yml
143   when: fp_ntpdate|bool
144   tags: fp_ntpdate
145
146 - name: Import nfs.yml
147   ansible.builtin.import_tasks: nfs.yml
148   when: fp_nfs|bool
149   tags: fp_nfs
150
151 - name: Import nfsd.yml
152   ansible.builtin.import_tasks: nfsd.yml
153   when: fp_nfsd|bool
154   tags: fp_nfsd
155
156 - name: Import smartd.yml
157   ansible.builtin.import_tasks: smartd.yml
158   when: fp_smartd|bool
159   tags: fp_smartd
160
161 - name: Import apcupsd.yml
162   ansible.builtin.import_tasks: apcupsd.yml
163   when: fp_apcupsd|bool
164   tags: fp_apcupsd
165
166 - name: Import hostapd.yml
167   ansible.builtin.import_tasks: hostapd.yml
168   when: fp_hostapd|bool
169   tags: fp_hostapd
170
171 - name: Import procmail.yml
172   ansible.builtin.import_tasks: procmail.yml
173   when: fp_procmail|bool
174   tags: fp_procmail
175
176 - name: Import motd.yml
177   ansible.builtin.import_tasks: motd.yml
178   when: fp_motd|bool
179   tags: fp_motd
180
181 - name: Import gitserver.yml
182   ansible.builtin.import_tasks: gitserver.yml
183   when: fp_gitserver|bool
184   tags: fp_gitserver
185
186 - name: Import sudoers.yml
```

(continues on next page)

(continued from previous page)

```
187  ansible.builtin.import_tasks: sudoers.yml
188  when: fp_sudoers|bool
189  tags: fp_sudoers
190
191  - name: Import cron.yml
192    ansible.builtin.import_tasks: cron.yml
193    when: fp_cron|bool
194    tags: fp_cron
195
196  - name: Import resolvconf.yml
197    ansible.builtin.import_tasks: resolvconf.yml
198    when: fp_resolvconf|bool
199    tags: fp_resolvconf
200
201  - name: Import aliases.yml
202    ansible.builtin.import_tasks: aliases.yml
203    when: fp_aliases|bool
204    tags: fp_aliases
205
206  - name: Import periodic.yml
207    ansible.builtin.import_tasks: periodic.yml
208    when: fp_periodic|bool
209    tags: fp_periodic
210
211  - name: Import wpasupplicant.yml
212    ansible.builtin.import_tasks: wpasupplicant.yml
213    when: fp_wpasupplicant|bool
214    tags: fp_wpasupplicant
215
216  - name: Import linux.yml
217    ansible.builtin.import_tasks: linux.yml
218    when: fp_linux|bool
219    tags: fp_linux
220
221  - name: Import login.yml
222    ansible.builtin.import_tasks: login.yml
223    when: fp_login|bool
224    tags: fp_login
225
226  - name: Import snmpd.yml
227    ansible.builtin.import_tasks: snmpd.yml
228    when: fp_snmpd|bool
229    tags: fp_snmpd
230
231  - name: Import qemu.yml
232    ansible.builtin.import_tasks: qemu.yml
233    when: fp_qemu|bool
234    tags: fp_qemu
235
236  # EOF
237  ...
```

3.1.2 aliases.yml

Synopsis: Configure aliases.

Description of the task.

[tasks/aliases.yml]

```

1 ---
2
3 - name: "aliases: Configure {{ fp_aliases_conf_file }}"
4   ansible.builtin.lineinfile:
5     dest: "{{ fp_aliases_conf_file }}"
6     regexp: '^\\s*{{ item.alias }}:(.*)$'
7     line: '{{ item.alias }}: {{ item.user }}'
8     backup: "{{ fp_backup }}"
9     loop: "{{ fp_aliases_conf }}"
10    notify: newaliases
11
12 # EOF
13 ...

```

3.1.3 apcupsd.yml

Synopsis: Configure apcupsd.

Description of the task.

[tasks/apcupsd.yml]

```

1 ---
2
3 - name: "apcupsd: Enable and start apcupsd"
4   ansible.builtin.lineinfile:
5     dest: /etc/rc.conf
6     regexp: '^\\s*apcupsd_enable\\s*=(.*)$'
7     line: 'apcupsd_enable="YES"'
8     backup: "{{ fp_backup }}"
9     when: fp_apcupsd_enabled|bool
10    notify: enable and start apcupsd
11    tags: fp_apcupsd_rcconf
12
13 - name: "apcupsd: Disable and stop apcupsd"
14   ansible.builtin.lineinfile:
15     dest: /etc/rc.conf
16     regexp: '^\\s*apcupsd_enable\\s*=(.*)$'
17     line: 'apcupsd_enable="NO"'
18     backup: "{{ fp_backup }}"
19     when: not fp_apcupsd_enabled|bool
20    notify: disable and stop apcupsd
21    tags: fp_apcupsd_rcconf
22
23 - name: "apcupsd: Create directory {{ fp_apcupsd_conf_dir }}"
24   ansible.builtin.file:

```

(continues on next page)

(continued from previous page)

```

25  state: directory
26  path: "{{ fp_apcupsd_conf_dir }}"
27  mode: "{{ fp_apcupsd_conf_dir_mode }}"
28  tags: fp_apcupsd_conf
29
30 - name: "apcupsd: Configure apcupsd.conf"
31  ansible.builtin.template:
32    src: apcupsd.conf.j2
33    dest: "{{ fp_apcupsd_conf_file }}"
34    owner: "{{ fp_apcupsd_conf_owner }}"
35    group: "{{ fp_apcupsd_conf_group }}"
36    mode: "{{ fp_apcupsd_conf_mode }}"
37    backup: "{{ fp_backup }}"
38  notify: restart apcupsd
39  tags: fp_apcupsd_conf
40
41 # EOF
42 ...

```

3.1.4 authorized-key.yml

Synopsis: Configure authorized-key.

Description of the task.

[tasks/authorized-key.yml]

```

1  ---
2
3  - name: "authorized-key: Get list of users"
4    ansible.builtin.getent:
5      database: passwd
6      tags: fp_authorized_key_debug
7
8  - name: "authorized-key: Debug (fp_authorized_key_debug={{ fp_authorized_key_debug }})"
9    vars:
10     msg: |-
11       fp_authorized_key_fail: {{ fp_authorized_key_fail|bool }}
12       fp_authorized_key_install_individually: {{ fp_authorized_key_install_
↪individually|bool }}
13       fp_authorized_key_install_exclusive: {{ fp_authorized_key_install_exclusive|bool }}
14       fp_authorized_key_follow: {{ fp_authorized_key_follow|bool }}
15       fp_authorized_key_key_options: {{ fp_authorized_key_key_options|default('UNDEFINED
↪') }}
16       fp_authorized_key_manage_dir: {{ fp_authorized_key_manage_dir|bool }}
17       fp_authorized_key_path: {{ fp_authorized_key_path|default('UNDEFINED') }}
18       fp_authorized_key_state: {{ fp_authorized_key_state|default('present') }}
19       fp_authorized_key_validate_certs: {{ fp_authorized_key_validate_certs|bool }}
20       fp_authorized_key_conf:
21         {{ fp_authorized_key_conf|to_nice_yaml|indent(2) }}
22       getent_passwd keys:
23         {{ getent_passwd.keys()|list|sort|to_yaml|indent(2) }}

```

(continues on next page)

(continued from previous page)

```

24  ansible.builtin.debug:
25    msg: "{{ ' {}'.format(msg) }}"
26  when: fp_authorized_key_debug|bool
27  tags: fp_authorized_key_debug
28
29  - name: "authorized-key: Configure authorized keys individually"
30  ansible.posix.authorized_key:
31    user: "{{ item.user }}"
32    key: "{{ item.key }}"
33    exclusive: "{{ item.exclusive|default(omit) }}"
34    comment: "{{ item.comment|default(omit) }}"
35    follow: "{{ item.follow|default(omit) }}"
36    key_options: "{{ item.key_options|default(omit) }}"
37    manage_dir: "{{ item.manage_dir|default(omit) }}"
38    path: "{{ item.path|default(omit) }}"
39    state: "{{ item.state|default(omit) }}"
40    validate_certs: "{{ item.validate_certs|default(omit) }}"
41  loop: "{{ fp_authorized_key_conf }}"
42  loop_control:
43    label: "{{ item.user }}"
44  when:
45    - fp_authorized_key_install_individually|bool
46    - fp_authorized_key_fail|bool or (item.user in getent_passwd.keys())
47
48  - name: "authorized-key: Configure joined authorized keys"
49  block:
50    - name: "authorized-key: Create list of users"
51      ansible.builtin.set_fact:
52        local_list_users: "{{ fp_authorized_key_conf|map(attribute='user')|unique }}"
53    - name: "authorized-key: Debug show list of users"
54      ansible.builtin.debug:
55        var: local_list_users
56      when: fp_authorized_key_debug|bool
57    - name: "authorized-key: Debug show joined keys"
58      ansible.builtin.debug:
59        msg: "{{ ' {}'.format(msg) }}"
60      loop: "{{ local_list_users }}"
61      vars:
62        my_query: "[?user=='{{ item }}'].key"
63        msg: "{{ fp_authorized_key_conf|json_query(my_query)|join('\n') }}"
64      when: fp_authorized_key_debug|bool
65    - name: "authorized-key: Configure joined authorized keys"
66      ansible.posix.authorized_key:
67        user: "{{ item }}"
68        key: "{{ fp_authorized_key_conf|json_query(my_query)|join('\n') }}"
69        exclusive: "{{ fp_authorized_key_install_exclusive|bool }}"
70        comment: "{{ fp_authorized_key_comment|default(omit) }}"
71        follow: "{{ fp_authorized_key_follow|bool }}"
72        key_options: "{{ fp_authorized_key_key_options|default(omit) }}"
73        manage_dir: "{{ fp_authorized_key_manage_dir|bool }}"
74        path: "{{ fp_authorized_key_path|default(omit) }}"
75        state: "{{ fp_authorized_key_state|default(omit) }}"

```

(continues on next page)

(continued from previous page)

```

76     validate_certs: "{{ fp_authorized_key_validate_certs|bool }}"
77     loop: "{{ local_list_users }}"
78     vars:
79         my_query: "[?user=='{{ item }}'].key"
80     when:
81         - not fp_authorized_key_install_individually|bool
82         - fp_authorized_key_fail|bool or (item.user in getent_passwd.keys())
83
84 # EOF
85 ...

```

3.1.5 cloud-providers.yml

Synopsis: Configure cloud-providers.

Description of the task.

[tasks/cloud-providers.yml]

```

1 ---
2
3 - name: Import cloud-providers/digitalocean.yml
4   import_tasks: cloud-providers/digitalocean.yml
5   when: digitalocean|bool
6   tags: fp_do
7
8 # EOF
9 ...

```

3.1.6 copy.yml

Synopsis: Configure copy.

Description of the task.

[tasks/copy.yml]

```

1 ---
2
3 - name: "copy: Debug (fp_copy_debug={{ fp_copy_debug }})"
4   vars:
5     msg: |-
6       fp_unarch_default: {{ fp_unarch_default }}
7       fp_unarch_optional: {{ fp_unarch_optional }}
8       fp_synchronize_default:
9         {{ fp_synchronize_default|to_nice_yaml|indent(2) }}
10      fp_synchronize_optional:
11        {{ fp_synchronize_optional|to_nice_yaml|indent(2) }}
12      fp_copy_dir: {{ fp_copy_dir }}
13      fp_copy_dir_default: {{ fp_copy_dir_default }}
14      fp_copy_arch_default: {{ fp_copy_arch_default }}
15      fp_copy_dir_optional: {{ fp_copy_dir_optional }}

```

(continues on next page)

(continued from previous page)

```

16     fp_copy_arch_optional: {{ fp_copy_arch_optional }}
17     fp_unarchive_keep_newer: {{ fp_unarchive_keep_newer }}
18     fp_synchronize_delete: {{ fp_synchronize_delete }}
19     fp_synchronize_archive: {{ fp_synchronize_archive }}
20     fp_synchronize_checksum: {{ fp_synchronize_checksum }}
21     fp_synchronize_compress: {{ fp_synchronize_compress }}
22     fp_synchronize_copy_links: {{ fp_synchronize_copy_links }}
23     fp_synchronize_dirs: {{ fp_synchronize_dirs }}
24     fp_synchronize_existing_only: {{ fp_synchronize_existing_only }}
25     fp_synchronize_group: {{ fp_synchronize_group }}
26     fp_synchronize_mode: {{ fp_synchronize_mode }}
27     fp_synchronize_owner: {{ fp_synchronize_owner }}
28     fp_synchronize_perms: {{ fp_synchronize_perms }}
29     fp_synchronize_times: {{ fp_synchronize_times }}
30     fp_synchronize_verify_host: {{ fp_synchronize_verify_host }}
31 ansible.builtin.debug:
32     msg: "{{ '{ }'.format(msg) }}"
33 when: fp_copy_debug|bool
34 tags: fp_copy_debug
35
36 # Sync default
37 - name: "copy: Synchronize default files {{ fp_copy_dir }}/{{ fp_copy_dir_default }}"
38   ansible.builtin.import_tasks: copy/sync-default.yml
39   when: fp_synchronize_default|length > 0
40   tags: fp_copy_sync_default
41
42 # Sync optional
43 - name: "copy: Synchronize optional files {{ fp_copy_dir }}/{{ fp_copy_dir_optional }}"
44   ansible.builtin.import_tasks: copy/sync-optional.yml
45   when: fp_synchronize_optional|length > 0
46   tags: fp_copy_sync_optional
47
48 # Unarch default
49 - name: "copy: Extract default files {{ fp_copy_dir }}/{{ fp_copy_arch_defaul }}"
50   ansible.builtin.import_tasks: copy/unarch-default.yml
51   when: fp_unarch_default|bool
52   tags: fp_copy_unarch_default
53
54 # Unarch optional
55 - name: "copy: Extract optional files {{ fp_copy_dir }}/{{ fp_copy_arch_optional }}"
56   ansible.builtin.import_tasks: copy/unarch-optional.yml
57   when: fp_unarch_optional|bool
58   tags: fp_copy_unarch_optional
59
60 # EOF
61 ...

```


3.1.7 cron.yml

Synopsis: Configure cron.

Description of the task.

[tasks/cron.yml]

```

1 ---
2
3 - name: "cron: Debug (fp_cron_debug={{ fp_cron_debug }})"
4   vars:
5     msg: |-
6       fp_cron_dir: {{ fp_cron_dir }}
7       fp_cron_var:
8         {{ fp_cron_var|to_yaml|indent(2) }}
9       fp_cron_tab:
10        {{ fp_cron_tab|to_nice_yaml|indent(2) }}
11       fp_cron_file_system: {{ fp_cron_file_system }}
12       fp_cron_var_system:
13         {{ fp_cron_var_system|to_yaml|indent(2) }}
14       fp_cron_tab_system:
15         {{ fp_cron_tab_system|to_nice_yaml|indent(2) }}
16   ansible.builtin.debug:
17     msg: "{{ '{ }'.format(msg) }}"
18   when: fp_cron_debug|bool
19   tags: fp_cron_debug
20
21   # Note on "cron - Manage cron.d and crontab entries"
22   # https://docs.ansible.com/ansible/latest/modules/cron_module.html
23   # Quoting from Synopsis: "When crontab jobs are managed: the module
24   # includes one line with the description of the crontab entry
25   # "#Ansible: <name>" corresponding to the "name" passed to the module,
26   # which is used by future ansible/module calls to find/check the
27   # state. The "name" parameter should be unique, and changing the
28   # "name" value will result in a new cron task being created (or a
29   # different one being removed).
30
31   # If "# <name>" entry is present change it to "#Ansible: <name>" to
32   # avoid double-entries
33
34 - name: "cron: Rename present crontab entries"
35   ansible.builtin.include_tasks:
36     file: fn/cron-rename-entry.yml
37     loop: "{{ fp_cron_tab }}"
38     tags: fp_cron_rename_entries
39
40 - name: "cron: Rename present crontab entries system"
41   ansible.builtin.replace:
42     path: "{{ fp_cron_file_system }}"
43     regexp: '^\\s*#\\s*{{ item.name }}$'
44     replace: "#Ansible: {{ item.name }}"
45     backup: "{{ fp_backup }}"
46     loop: "{{ fp_cron_tab }}"

```

(continues on next page)

(continued from previous page)

```

47  loop_control:
48      label: "{{ item.name }}"
49  tags: fp_cron_rename_entries_system
50
51 - name: "cron: Configure cron variables"
52  community.general.cronvar:
53      name: "{{ item.name }}"
54      value: "{{ item.value }}"
55      user: "{{ item.user }}"
56  loop: "{{ fp_cron_var }}"
57  tags: fp_cron_var
58
59 - name: "cron: Configure crontab entries"
60  ansible.builtin.cron:
61      state: "{{ item.state }}"
62      user: "{{ item.user }}"
63      name: "{{ item.name }}"
64      minute: "{{ item.minute }}"
65      hour: "{{ item.hour }}"
66      day: "{{ item.day }}"
67      month: "{{ item.month }}"
68      weekday: "{{ item.weekday }}"
69      job: "{{ item.command }}"
70  loop: "{{ fp_cron_tab }}"
71  loop_control:
72      label: "{{ item.name }}"
73  tags: fp_cron_tab
74
75 - name: "cron: Configure cron variables system"
76  community.general.cronvar:
77      cron_file: "{{ fp_cron_file_system }}"
78      name: "{{ item.name }}"
79      value: "{{ item.value }}"
80  loop: "{{ fp_cron_var_system }}"
81  tags: fp_cron_var_system
82
83 - name: "cron: Configure crontab entries system"
84  ansible.builtin.cron:
85      cron_file: "{{ fp_cron_file_system }}"
86      state: "{{ item.state }}"
87      user: "{{ item.user }}"
88      name: "{{ item.name }}"
89      minute: "{{ item.minute }}"
90      hour: "{{ item.hour }}"
91      day: "{{ item.day }}"
92      month: "{{ item.month }}"
93      weekday: "{{ item.weekday }}"
94      job: "{{ item.command }}"
95  loop: "{{ fp_cron_tab_system }}"
96  loop_control:
97      label: "{{ item.name }}"
98  tags: fp_cron_tab_system

```

(continues on next page)

(continued from previous page)

```

99
100 # TODO:
101 # /etc/cron.d
102 # /usr/local/etc/cron.d
103
104 # EOF
105 ...

```

3.1.8 debug.yml

Synopsis: Configure debug.

Description of the task.

[tasks/debug.yml]

```

1 ---
2
3 - name: "FreeBSD postinstall Debug"
4   vars:
5     _fp_qemu_install_varnames: "{{ query('varnames', '^fp_qemu_(.*)install$') }}"
6     _fp_qemu_install_values: "{{ _fp_qemu_install_varnames|map('extract', vars)|list }}"
7     _fp_qemu_install_dict: "{{ dict(_fp_qemu_install_varnames|zip(_fp_qemu_install_
8 ↪values)) }}"
9     msg: |-
10      ansible_architecture: {{ ansible_architecture }}
11      ansible_os_family: {{ ansible_os_family }}
12      ansible_distribution: {{ ansible_distribution }}
13      ansible_distribution_major_version: {{ ansible_distribution_major_version }}
14      ansible_distribution_version: {{ ansible_distribution_version }}
15      ansible_distribution_release: {{ ansible_distribution_release }}
16
17      rc_conf_hostname: {{ rc_conf_hostname }}
18      fp_domain: {{ fp_domain }}
19      aws: {{ aws }}
20      digitalocean: {{ digitalocean }}
21      gcp: {{ gcp }}
22
23      freebsd_install_method: {{ freebsd_install_method }}
24      freebsd_use_packages: {{ freebsd_use_packages|default('UNDEFINED') }}
25      freebsd_install_retries: {{ freebsd_install_retries }}
26      freebsd_install_delay: {{ freebsd_install_delay }}
27
28      fp_install: {{ fp_install }}
29      fp_backup: {{ fp_backup }}
30      fp_sanity: {{ fp_sanity }} fp_sanity_tags_fatal: {{ fp_sanity_tags_fatal }}
31
32      fp_aliases: {{ fp_aliases }}
33      fp_apcupsd: {{ fp_apcupsd }} fp_apcupsd_install: {{ fp_apcupsd_install }} fp_
34 ↪apcupsd_enabled: {{ fp_apcupsd_enabled }}
35      fp_authorized_key: {{ fp_authorized_key }}
36      fp_copy: {{ fp_copy }} fp_unarch_default: {{ fp_unarch_default }} fp_unarch_
37 ↪optional: {{ fp_unarch_optional }}

```

(continues on next page)

(continued from previous page)

```

35     fp_cron: {{ fp_cron }}
36     fp_devfs: {{ fp_devfs }} fp_devfs_enabled: {{ fp_devfs_enabled }}
37     fp_fstab: {{ fp_fstab }}
38     fp_gitserver: {{ fp_gitserver }} fp_gitserver_install: {{ fp_gitserver_install }}
↪}
39     fp_groups: {{ fp_groups }}
40     fp_groupwrappers_install: {{ fp_groupwrappers_install }}
41     fp_hostapd: {{ fp_hostapd }} fp_hostapd_install: {{ fp_hostapd_install }} fp_
↪hostapd_enabled: {{ fp_hostapd_enabled }}
42     fp_hostname: {{ fp_hostname }}
43     fp_linux: {{ fp_linux }} fp_linux_install: {{ fp_linux_install }} fp_linux_
↪enable: {{ fp_linux_enable }}
44     fp_loader: {{ fp_loader }} fp_loader_warning: {{ fp_loader_warning }}
45     fp_login: {{ fp_login }} fp_login_conf_global: {{ fp_login_conf_global }}
46     fp_make: {{ fp_make }}
47     fp_motd: {{ fp_motd }}
48     fp_mount_img: {{ fp_mount_img }}
49     fp_mount_iso: {{ fp_mount_iso }}
50     fp_nfs: {{ fp_nfs }} fp_nfs_enable: {{ fp_nfs_enable }}
51     fp_nfsd: {{ fp_nfsd }} fp_nfsd_enable: {{ fp_nfsd_enable }}
52     fp_ntp: {{ fp_ntp }} fp_ntp_enable: {{ fp_ntp_enable }}
53     fp_ntpdate: {{ fp_ntpdate }}
54     fp_passwords: {{ fp_passwords }}
55     fp_periodic: {{ fp_periodic }}
56     fp_procmail: {{ fp_procmail }} fp_procmail_install: {{ fp_procmail_install }}
57     fp_qemu: {{ fp_qemu }} fp_qemu_enabled: {{ fp_qemu_enabled }}{% for k,v in _fp_
↪qemu_install_dict.items() %} {{ k }}: {{ v }}{% endfor %}
58
59     fp_resolvconf: {{ fp_resolvconf }}
60     fp_smartd: {{ fp_smartd }} fp_smartd_install: {{ fp_smartd_install }} fp_smartd_
↪enabled: {{ fp_smartd_enabled }}
61     fp_snmpd: {{ fp_snmpd }} fp_snmpd_install: {{ fp_snmpd_install }} fp_snmpd_
↪enabled: {{ fp_snmpd_enabled }}
62     fp_ssh: {{ fp_ssh }}
63     fp_sshd: {{ fp_sshd }} fp_sshd_enable: {{ fp_sshd_enable }}
64     fp_sudoers: {{ fp_sudoers }}
65     fp_swap: {{ fp_swap }} fp_swap_enable: {{ fp_swap_enable }}
66     fp_sysctl: {{ fp_sysctl }}
67     fp_timezone: {{ fp_timezone }}
68     fp_tmpmfs: {{ fp_tmpmfs }}
69     fp_users: {{ fp_users }}
70     fp_wpasuppliant: {{ fp_wpasuppliant }} fp_wpasuppliant_install: {{ fp_
↪wpasuppliant_install }}
71
72     fp_mountd_enable: {{ fp_mountd_enable }}
73     fp_rpcbind_enable: {{ fp_rpcbind_enable }}
74     fp_rpc_lockd_enable: {{ fp_rpc_lockd_enable }}
75     fp_rpc_statd_enable: {{ fp_rpc_statd_enable }}
76     ansible.builtin.debug:
77         msg: "{{ ' {}'.format(msg) }}"
78
79 # EOF

```

(continues on next page)

(continued from previous page)

...

3.1.9 devfs.yml

Synopsis: Configure devfs.

Description of the task.

[tasks/devfs.yml]

```

1  ---
2
3  - name: "devfs: Debug (fp_devfs_debug={{ fp_devfs_debug }})"
4    vars:
5      msg: |-
6          fp_devfs_enabled: {{ fp_devfs_enabled }}
7          fp_devfs_load_rulesets: {{ fp_devfs_load_rulesets }}
8          fp_devfs_system_ruleset: {{ fp_devfs_system_ruleset }}
9          fp_devfs_rulesets: {{ fp_devfs_rulesets }}
10         fp_devfs_set_rulesets: {{ fp_devfs_set_rulesets }}
11         fp_devfs_rcconf:
12             {{ fp_devfs_rcconf|to_yaml|indent(2) }}
13         fp_devfs_conf_path: {{ fp_devfs_conf_path }}
14         fp_devfs_conf:
15             {{ fp_devfs_conf|to_yaml|indent(2) }}
16         fp_devfs_rules_path: {{ fp_devfs_rules_path }}
17         fp_devfs_rules_mode: {{ fp_devfs_rules_mode }}
18         fp_devfs_rules:
19             {{ fp_devfs_rules|to_yaml|indent(2) }}
20     ansible.builtin.debug:
21         msg: "{{ '}'}.format(msg) }}"
22     when: fp_devfs_debug|bool
23     tags: fp_devfs_debug
24
25 - name: "devfs: Shell: /etc/rc.d/devfs rcvar | grep ^devfs*"
26   ansible.builtin.shell:
27     cmd: >
28         /etc/rc.d/devfs rcvar | grep ^devfs*
29   register: result
30   changed_when: false
31   ignore_errors: true
32   tags: fp_devfs_rcvar
33
34 - name: "devfs: Test rcvar for system module"
35   block:
36     - name: "devfs: ERROR no rcvar"
37       ansible.builtin.debug:
38         msg: |
39             [ERROR] unable to determine rcvar. End of play.
40             Variable rcvar is missing in the file /etc/rc.d/devfs
41             Put the following line into the file /etc/rc.d/devfs
42             rcvar="devfs_load_rulesets"

```

(continues on next page)

(continued from previous page)

```

43 - name: "devfs: Block: End of play."
44     ansible.builtin.meta: end_play
45 when:
46     - not ansible_check_mode
47     - result|default({}) is failed
48 tags: fp_devfs_rcvar
49
50 - name: "devfs: Debug print rcvar"
51     ansible.builtin.debug:
52         var: result.stdout_lines|default('not defined')
53     when: fp_devfs_debug|bool
54     tags: fp_devfs_rcvar
55
56 - name: "devfs: Enable and start devfs"
57     ansible.builtin.lineinfile:
58         dest: /etc/rc.conf
59         regexp: '^\\s*devfs_load_rulesets\\s*=(.*)$'
60         line: 'devfs_load_rulesets="YES"'
61         backup: "{{ fp_backup }}"
62     notify: enable and start devfs
63     when: fp_devfs_enabled|bool
64     tags: fp_devfs_rcconf
65
66 - name: "devfs: Disable and stop devfs"
67     ansible.builtin.lineinfile:
68         dest: /etc/rc.conf
69         regexp: '^\\s*devfs_load_rulesets\\s*=(.*)$'
70         line: 'devfs_load_rulesets="NO"'
71         backup: "{{ fp_backup }}"
72     notify: disable and stop devfs
73     when: not fp_devfs_enabled|bool
74     tags: fp_devfs_rcconf
75
76 - name: "devfs: Configure devfs in /etc/rc.conf"
77     ansible.builtin.lineinfile:
78         dest: /etc/rc.conf
79         regexp: '^\\s*{{ item.key }}\\s*=(.*)$'
80         line: '{{ item.key }}="{{ item.value }}"'
81         backup: "{{ fp_backup }}"
82     loop: "{{ fp_devfs_rcconf }}"
83     notify: restart devfs
84     tags: fp_devfs_rcconf
85
86 - name: "devfs: Configure {{ fp_devfs_conf_path }}"
87     ansible.builtin.lineinfile:
88         dest: "{{ fp_devfs_conf_path }}"
89         regexp: '^\\s*{{ item.action }}\\s+{{ item.devname }}\\s+{{ item.arg }}'
90         line: "{{ item.action }} {{ item.devname }} {{ item.arg }}"
91         backup: "{{ fp_backup }}"
92     loop: "{{ fp_devfs_conf }}"
93     notify: restart devfs
94     tags: fp_devfs_conf

```

(continues on next page)

(continued from previous page)

```

95
96 - name: "devfs: Configure {{ fp_devfs_rules_path }}"
97   community.general.ini_file:
98     path: "{{ fp_devfs_rules_path }}"
99     create: true
100    mode: "{{ fp_devfs_rules_mode }}"
101    allow_no_value: true
102    owner: "{{ fp_devfs_owner }}"
103    group: "{{ fp_devfs_group }}"
104    section: "{{ item.section }}"
105    option: "{{ item.option }}"
106    value: "{{ item.value|default(omit) }}"
107    state: "{{ item.state|(default('present')) }}"
108    backup: "{{ fp_backup }}"
109    loop: "{{ fp_devfs_rules }}"
110    notify: restart devfs
111    tags: fp_devfs_rules
112
113 # EOF
114 ...

```

3.1.10 dhclient.yml

Synopsis: Configure dhclient.

Description of the task.

[tasks/dhclient.yml]

```

1 ---
2
3 - name: "dhclient: Debug (fp_dhclient_debug={{ fp_dhclient_debug }})"
4   vars:
5     msg: |-
6       fp_dhclient_ifn: {{ fp_dhclient_ifn }}
7       fp_dhclient_ifn_rcconf: {{ fp_dhclient_ifn_rcconf }}
8       fp_dhclient_ifn_leases: {{ fp_dhclient_ifn_leases }}
9       fp_dhclient_sanity: {{ fp_dhclient_sanity }}
10      fp_dhclient_conf_path: {{ fp_dhclient_conf_path }}
11
12      fp_dhclient_conf_defaults:
13        {{ fp_dhclient_conf_defaults|to_yaml|indent(2) }}
14      fp_dhclient_conf_blocks:
15        {{ fp_dhclient_conf_blocks|to_nice_yaml|indent(2) }}
16    ansible.builtin.debug:
17      msg: "{{ '{ }'.format(msg) }}"
18    when: fp_dhclient_debug|bool
19    tags: fp_dhclient_debug
20
21 - name: "dhclient: Sanity"
22   block:
23     - name: "dhclient: Sanity: List fp_dhclient_ifn empty"

```

(continues on next page)

(continued from previous page)

```

24  ansible.builtin.assert:
25    that: (fp_dhclient_ifn|length == 0)|
26      ternary(fp_dhclient_ifn_rconf != fp_dhclient_ifn_leases, true)
27    fail_msg: |-
28      The list of interfaces fp_dhclient_ifn is empty. Either
29      fp_dhclient_ifn_rconf or fp_dhclient_ifn_leases must be True,
30      but not both.
31  - name: "dhclient: Sanity: List fp_dhclient_ifn not empty"
32  ansible.builtin.assert:
33    that: (fp_dhclient_ifn|length > 0)|
34      ternary(not(fp_dhclient_ifn_rconf or fp_dhclient_ifn_leases), true)
35    fail_msg: |-
36      The list of interfaces fp_dhclient_ifn is not empty. Both
37      fp_dhclient_ifn_rconf and fp_dhclient_ifn_leases must be
38      False.
39  when: fp_dhclient_sanity|bool
40  tags: fp_dhclient_sanity
41
42  - name: Get interfaces from /var/db/dhclient.leases.*
43  block:
44    - name: "dhclient: Get interfaces /var/db/dhclient.leases.*"
45      ansible.builtin.shell:
46        cmd: ls -1 /var/db/dhclient* | rev | cut -d . -f 1 | rev
47      register: result
48      changed_when: false
49    - name: "dhclient: Debug interfaces /var/db/dhclient.leases.*"
50      ansible.builtin.debug:
51        var: result.stdout_lines
52      when: fp_dhclient_debug|bool
53    - name: "dhclient: Get status of dhclients"
54      ansible.builtin.command:
55        cmd: "/etc/rc.d/dhclient status {{ item }}"
56      register: result
57      failed_when: false
58      changed_when: false
59      loop: "{{ result.stdout_lines }}"
60    - name: "dhclient: Set fp_dhclient_ifn"
61      ansible.builtin.set_fact:
62        fp_dhclient_ifn: "{{ result.results|
63          rejectattr('rc', 'gt', 0)|
64          map(attribute='item')|list }}"
65    - name: "dhclient: Debug fp_dhclient_ifn"
66      ansible.builtin.debug:
67        var: fp_dhclient_ifn
68      when: fp_dhclient_debug|bool
69  when: fp_dhclient_ifn_leases|bool
70  tags: fp_dhclient_ifn_leases
71
72  - name: Get interfaces from /etc/rc.conf
73  block:
74    - name: "dhclient: Get interfaces from /etc/rc.conf"
75      ansible.builtin.command:

```

(continues on next page)

(continued from previous page)

```

76     cmd: grep '^ifconfig_.*$' /etc/rc.conf
77     register: result
78     changed_when: false
79     - name: "dhclient: Debug interfaces from /etc/rc.conf"
80     ansible.builtin.debug:
81         var: result.stdout_lines
82     when: fp_dhclient_debug|bool
83     - name: "dhclient: Set fp_dhclient_ifn"
84     ansible.builtin.set_fact:
85         fp_dhclient_ifn: "{{ _keys }}"
86     vars:
87         _regex: '^ifconfig_.*\s*=.*DHCP.*$'
88         _arr: "{{ result.stdout_lines|
89                 select('match', _regex)|
90                 map('split', '=')|list }}"
91         _keys: "{{ _arr|
92                 map(attribute=0)|
93                 map('split', '_')|
94                 map(attribute=1)|list }}"
95     - name: "dhclient: Debug fp_dhclient_ifn"
96     ansible.builtin.debug:
97         var: fp_dhclient_ifn
98     when: fp_dhclient_debug|bool
99 when: fp_dhclient_ifn_rcconf|bool
100 tags: fp_dhclient_ifn_rcconf
101
102 - name: "dhclient: Create {{ fp_dhclient_conf_path }}"
103 ansible.builtin.file:
104     path: "{{ fp_dhclient_conf_path }}"
105     owner: root
106     group: wheel
107     mode: '0644'
108 tags: fp_dhclient_conf_create
109
110 - name: "dhclient: Configure defaults in {{ fp_dhclient_conf_path }}"
111 ansible.builtin.lineinfile:
112     dest: "{{ fp_dhclient_conf_path }}"
113     insertbefore: BOF
114     regexp: '{{ item.key }}\s+(.*)$'
115     line: '{{ item.key }} {{ item.value }};'
116     backup: "{{ fp_backup }}"
117     loop: "{{ fp_dhclient_conf_defaults }}"
118     notify: restart dhclient all
119     tags: fp_dhclient_conf_defaults
120
121 - name: "dhclient: Configure blocks in {{ fp_dhclient_conf_path }}"
122 ansible.builtin.blockinfile:
123     dest: "{{ fp_dhclient_conf_path }}"
124     insertafter: EOF
125     marker: "# {mark} {{ item.type }} {{ item.ifn }}"
126     block: "{{ item.block }}"
127     backup: "{{ fp_backup }}"

```

(continues on next page)

(continued from previous page)

```

128 register: fp_dhclient_conf_changes
129 loop: "{{ fp_dhclient_conf_blocks }}"
130 loop_control:
131     label: "{{ item.type }} {{ item.ifn }}"
132 notify: restart dhclient blocks
133 tags: fp_dhclient_conf_blocks
134
135 - name: "dhclient: Debug changed interfaces to be restarted"
136   ansible.builtin.debug:
137     msg: "{{ fp_dhclient_conf_changes.results|
138           selectattr('changed')|
139           map(attribute='item.ifn')|
140           unique }}"
141   when: fp_dhclient_debug|bool
142   tags: fp_dhclient_conf_blocks
143
144 # EOF
145 ...

```

3.1.11 fstab.yml

Synopsis: Configure fstab.

Description of the task.

[tasks/fstab.yml]

```

1 ---
2
3 - name: "fstab: Configure fstab entries"
4   ansible.posix.mount:
5     name: "{{ item.name }}"
6     src: "{{ item.src }}"
7     fstype: "{{ item.fstype }}"
8     opts: "{{ item.opts }}"
9     dump: "{{ item.dump|default('0') }}"
10    passno: "{{ item.passno|default('0') }}"
11    state: "{{ item.state|default('mounted') }}"
12    backup: "{{ fp_backup }}"
13    loop: "{{ fp_fstab_entries }}"
14
15 # EOF
16 ...

```

3.1.12 gitserver.yml

Synopsis: Configure gitserver.

Description of the task.

[tasks/gitserver.yml]

```

1 ---
2
3 - name: "gitserver: Add {{ fp_gitserver_group }} group"
4   ansible.builtin.group:
5     name: "{{ fp_gitserver_group }}"
6     gid: "{{ fp_gitserver_group_gid|default(omit) }}"
7     tags: fp_gitserver_group
8
9 - name: "gitserver: Add {{ fp_gitserver_owner }} user"
10  ansible.builtin.user:
11    name: "{{ fp_gitserver_owner }}"
12    uid: "{{ fp_gitserver_owner_uid|default(omit) }}"
13    shell: "{{ fp_gitserver_owner_shell|default(omit) }}"
14    home: "{{ fp_gitserver_owner_home|default(omit) }}"
15    tags: fp_gitserver_owner
16
17 - name: "gitserver: Add user {{ fp_gitserver_owner }} to {{ fp_gitserver_group }} group"
18  ansible.builtin.user:
19    name: "{{ fp_gitserver_owner }}"
20    groups: "{{ fp_gitserver_group }}"
21    tags: fp_gitserver_owner
22
23 - name: "gitserver: Set ownership and permissions of {{ fp_gitserver_dir }}"
24  ansible.builtin.file:
25    state: directory
26    path: "{{ fp_gitserver_dir }}"
27    owner: "{{ fp_gitserver_owner }}"
28    group: "{{ fp_gitserver_group }}"
29    mode: "{{ fp_gitserver_dir_mode }}"
30    tags: fp_gitserver_dir
31
32 # TODO: How do I Setup up a Multi-User Git Server
33 # https://forums.freebsd.org/threads/how-do-i-setup-up-a-multi-user-git-server.50969/
34
35 # EOF
36 ...

```

3.1.13 groups.yml

Synopsis: Configure groups.

Description of the task.

[tasks/groups.yml]

```

1 ---
2
3 - name: "groups: Debug (fp_groups_debug={{ fp_groups_debug }})"
4   vars:
5     msg: |-
6       fp_groups_conf:
7         {{ fp_groups_conf|to_nice_yaml|indent(2) }}
8     ansible.builtin.debug:
9       msg: "{{ '{ }'.format(msg) }}"
10    when: fp_groups_debug|bool
11    tags: fp_groups_debug
12
13 - name: "groups: Manage groups"
14   ansible.builtin.group:
15     name: "{{ item.name }}"
16     gid: "{{ item.gid|default(omit) }}"
17     local: "{{ item.local|default(omit) }}"
18     state: "{{ item.state|default(omit) }}"
19     system: "{{ item.system|default(omit) }}"
20     loop: "{{ fp_groups_conf }}"
21     tags: fp_groups_conf
22
23 # EOF
24 ...

```

3.1.14 groupwrappers.yml

Synopsis: Configure groupwrappers.

Description of the task.

[tasks/groupwrappers.yml]

```

1 ---
2
3 # Ansible module users needs utilities groupadd, groupdel, groupmod. There are
4 # no such utilities in BSD. Let us create them.
5
6 - name: "groupwrappers: Create directory for wrappers {{ fp_groupwrappers_dir }}"
7   ansible.builtin.file:
8     state: directory
9     path: "{{ fp_groupwrappers_dir }}"
10    mode: "{{ fp_groupwrappers_dir_mode }}"
11    owner: "{{ fp_groupwrappers_owner }}"
12    group: "{{ fp_groupwrappers_group }}"
13    tags: fp_groupwrappers_dir

```

(continues on next page)

(continued from previous page)

```

14
15 - name: "groupwrappers: Create groupadd wrapper in {{ fp_groupwrappers_dir }}"
16   ansible.builtin.template:
17     src: "groupadd.j2"
18     dest: "{{ fp_groupwrappers_dir }}/groupadd"
19     owner: "{{ fp_groupwrappers_owner }}"
20     group: "{{ fp_groupwrappers_group }}"
21     mode: "{{ fp_groupwrappers_mode }}"
22     backup: "{{ fp_backup }}"
23   tags: fp_groupwrappers_groupadd
24
25 - name: "groupwrappers: Create groupdel wrapper in {{ fp_groupwrappers_dir }}"
26   ansible.builtin.template:
27     src: "groupdel.j2"
28     dest: "{{ fp_groupwrappers_dir }}/groupdel"
29     owner: "{{ fp_groupwrappers_owner }}"
30     group: "{{ fp_groupwrappers_group }}"
31     mode: "{{ fp_groupwrappers_mode }}"
32     backup: "{{ fp_backup }}"
33   tags: fp_groupwrappers_groupdel
34
35 - name: "groupwrappers: Create groupmod wrapper in {{ fp_groupwrappers_dir }}"
36   ansible.builtin.template:
37     src: "groupmod.j2"
38     dest: "{{ fp_groupwrappers_dir }}/groupmod"
39     owner: "{{ fp_groupwrappers_owner }}"
40     group: "{{ fp_groupwrappers_group }}"
41     mode: "{{ fp_groupwrappers_mode }}"
42     backup: "{{ fp_backup }}"
43   tags: fp_groupwrappers_groupmod
44
45 # EOF
46 ...

```

3.1.15 hostapd.yml

Synopsis: Configure hostapd.

Description of the task.

[tasks/hostapd.yml]

```

1 ---
2
3 - name: "hostapd: Enable and start hostapd"
4   ansible.builtin.lineinfile:
5     dest: /etc/rc.conf
6     regexp: '^s*hostapd_enable\s*=(.*)$'
7     line: 'hostapd_enable="YES"'
8     backup: "{{ fp_backup }}"
9   notify: enable and start hostapd
10  when: fp_hostapd_enabled|bool

```

(continues on next page)

(continued from previous page)

```

11  tags: fp_hostapd_rcconf
12
13  - name: "hostapd: Disable and stop hostapd"
14    ansible.builtin.lineinfile:
15      dest: /etc/rc.conf
16      regexp: '^\\s*hostapd_enable\\s*=(.*)$'
17      line: 'hostapd_enable="NO"'
18      backup: "{{ fp_backup }}"
19    notify: disable and stop hostapd
20    when: not fp_hostapd_enabled|bool
21    tags: fp_hostapd_rcconf
22
23  - name: "hostapd: Configure interface for hostapd"
24    ansible.builtin.lineinfile:
25      dest: /etc/rc.conf
26      regexp: '^\\s*{{ item.key }}\\s*=(.*)$'
27      line: '{{ item.key }}="{{ item.value }}"'
28      backup: "{{ fp_backup }}"
29    loop: "{{ fp_hostapd_rcconf }}"
30    notify: restart hostapd
31    tags: fp_hostapd_rcconf
32
33  - name: "hostapd: Configure {{ fp_hostapd_conf_file }}"
34    ansible.builtin.template:
35      src: "{{ fp_hostapd_conf_template }}"
36      dest: "{{ fp_hostapd_conf_file }}"
37      owner: "{{ fp_hostapd_conf_owner }}"
38      group: "{{ fp_hostapd_conf_group }}"
39      mode: "{{ fp_hostapd_conf_mode }}"
40      backup: "{{ fp_backup }}"
41    notify: restart hostapd
42    tags: fp_hostapd_conf
43
44  # EOF
45  ...

```

3.1.16 hostname.yml

Synopsis: Configure hostname.

Description of the task.

[tasks/hostname.yml]

```

1  ---
2
3  - name: "hostname: Configure hostname in /etc/rc.conf"
4    ansible.builtin.lineinfile:
5      dest: /etc/rc.conf
6      regexp: '^\\s*{{ item.key }}\\s*=(.*)$'
7      line: '{{ item.key }}="{{ item.value }}"'
8      backup: "{{ fp_backup }}"

```

(continues on next page)

(continued from previous page)

```

9  loop: "{{ fp_hostname_rcconf }}"
10 notify: set hostname
11
12 # EOF
13 ...

```

3.1.17 hosts.yml

Synopsis: Configure hosts.

Description of the task.

[tasks/hosts.yml]

```

1  ---
2
3  - name: "hosts: Debug (fp_hosts_debug={{ fp_hosts_debug }})"
4    vars:
5      msg: |-
6        fp_hosts_sanity: {{ fp_hosts_sanity }}
7        fp_hosts_conf_file: {{ fp_hosts_conf_file }}
8        fp_hosts_conf_template: {{ fp_hosts_conf_template }}
9        fp_hosts_localhost_ipv4: {{ fp_hosts_localhost_ipv4 }}
10       fp_hosts_localhost_ipv6: {{ fp_hosts_localhost_ipv6 }}
11       fp_hosts_conf:
12         {{ fp_hosts_conf|to_yaml|indent(2) }}
13     ansible.builtin.debug:
14       msg: "{{ '{ }'.format(msg) }}"
15     when: fp_hosts_debug|bool
16     tags: fp_hosts_debug
17
18  - name: "Sanity"
19     block:
20       - name: "hosts: Sanity valid IP"
21         ansible.builtin.fail:
22           msg: "[ERROR] Invalid IP address"
23         when: (fp_hosts_conf|map(attribute='ip')|map('ipaddr')) is not all
24       - name: "hosts: Sanity unique FQDN"
25         ansible.builtin.fail:
26           msg: "[ERROR] FQDN not unique"
27         vars:
28           no_records: "{{ fp_hosts_conf|length }}"
29           no_hosts: "{{ fp_hosts_conf|map(attribute='fqdn')|list|unique|length }}"
30         when: no_records != no_hosts
31     when: fp_hosts_sanity|bool
32     tags: fp_hosts_sanity
33
34  - name: "hosts: Configure hosts in {{ fp_hosts_conf_file }}"
35     ansible.builtin.template:
36       src: "{{ fp_hosts_conf_template }}"
37       dest: "{{ fp_hosts_conf_file }}"
38       owner: root

```

(continues on next page)

(continued from previous page)

```

39  group: wheel
40  mode: "0644"
41  backup: "{{ fp_backup }}"
42  tags: fp_hosts_conf
43
44  # EOF
45  ...

```

3.1.18 libmap.yml

Synopsis: Configure libmap.

Description of the task.

[tasks/libmap.yml]

```

1  ---
2
3  - name: "libmap: Configure {{ fp_libmap_conf_file }}"
4    ansible.builtin.lineinfile:
5      dest: "{{ fp_libmap_conf_file }}"
6      regexp: '^s*{{ item.origin }}\s+(.*)$'
7      line: "{{ item.origin }} {{ item.target }}"
8      backup: "{{ fp_backup }}"
9      loop: "{{ fp_libmap_entries }}"
10
11  # TODO: includedir /usr/local/etc/libmap.d
12
13  # EOF
14  ...

```

3.1.19 linux.yml

Synopsis: Configure linux.

Description of the task.

[tasks/linux.yml]

```

1  ---
2
3  - name: "linux: Debug (fp_linux_debug={{ fp_linux_debug }})"
4    vars:
5      msg: |-
6        fp_linux_install: {{ fp_linux_install }}
7        fp_linux_enable: {{ fp_linux_enable }}
8        fp_linux_fstab:
9          {{ fp_linux_fstab|to_yaml|indent(2) }}
10       fp_linux_lib_dir: {{ fp_linux_lib_dir }}
11       fp_linux_lib:
12         {{ fp_linux_lib|to_yaml|indent(2) }}
13    ansible.builtin.debug:

```

(continues on next page)

(continued from previous page)

```

14     msg: "{{ '{' }.format(msg) }}"
15 when: fp_linux_debug|bool
16 tags: fp_linux_debug
17
18 - name: "linux: Enable linux"
19   ansible.builtin.lineinfile:
20     dest: /etc/rc.conf
21     regexp: '^\\s*linux_enable\\s*=(.*)$'
22     line: 'linux_enable="YES"'
23     backup: "{{ fp_backup }}"
24   when: fp_linux_enable|bool
25   tags: fp_linux_rcconf
26
27 - name: "linux: Disable linux"
28   ansible.builtin.lineinfile:
29     dest: /etc/rc.conf
30     regexp: '^\\s*linux_enable\\s*=(.*)$'
31     line: 'linux_enable="NO"'
32     backup: "{{ fp_backup }}"
33   when: not fp_linux_enable|bool
34   tags: fp_linux_rcconf
35
36 - name: "linux: Configure fstab"
37   ansible.posix.mount:
38     name: "{{ item.name }}"
39     src: "{{ item.src }}"
40     fstype: "{{ item.fstype }}"
41     opts: "{{ item.opts }}"
42     state: mounted
43   loop: "{{ fp_linux_fstab }}"
44   tags: fp_linux_fstab
45
46 - name: "linux: Create dirs for libraries"
47   ansible.builtin.file:
48     state: directory
49     path: "/compat/linux{{ item.src|dirname }}"
50     mode: "{{ item.dirmode|default('0755') }}"
51   loop: "{{ fp_linux_lib }}"
52   tags: fp_linux_lib_dir
53
54 - name: "linux: Copy libraries from {{ fp_linux_lib_dir }} to /compat/linux"
55   ansible.builtin.copy:
56     src: "{{ fp_linux_lib_dir }}{{ item.src }}"
57     dest: "/compat/linux{{ item.src }}"
58     mode: preserve
59   loop: "{{ fp_linux_lib }}"
60   tags: fp_linux_lib_copy
61
62 - name: "linux: Link libraries"
63   ansible.builtin.file:
64     state: link
65     src: "/compat/linux{{ item.src }}"

```

(continues on next page)

(continued from previous page)

```

66     dest: "/compat/linux{{ item.link }}"
67     loop: "{{ fp_linux_lib }}"
68     tags: fp_linux_lib_link
69
70 # EOF
71 ...

```

3.1.20 loader.yml

Synopsis: Configure loader.

Description of the task.

[tasks/loader.yml]

```

1 ---
2
3 - name: "loader: Debug (fp_loader_debug={{ fp_loader_debug }})"
4   vars:
5     msg: |-
6       fp_loader_warning: {{ fp_loader_warning }}
7       fp_loader_conf_template_enable: {{ fp_loader_conf_template_enable }}
8       fp_loader_load_modules: {{ fp_loader_load_modules }}
9       fp_loader_conf_file: {{ fp_loader_conf_file }}
10      fp_loader_conf_template: {{ fp_loader_conf_template }}
11      fp_loader_conf:
12        {{ fp_loader_conf|to_yaml|indent(2) }}
13      fp_loader_conf_modules:
14        {{ fp_loader_conf_modules|to_yaml|indent(2) }}
15      fp_sysctl_conf:
16        {{ fp_sysctl_conf|to_yaml|indent(2) }}
17    ansible.builtin.debug:
18      msg: "{{ '{ }'.format(msg) }}"
19    when: fp_loader_debug|bool
20    tags: fp_loader_debug
21
22 - name: "loader: Register sysctl variables that are settable via loader"
23   ansible.builtin.command:
24     cmd: sysctl -aNT
25   register: tmp_loader_vars
26   changed_when: false
27
28 - name: "loader: Create empty list fp_sysctl_loader_conf"
29   ansible.builtin.set_fact:
30     fp_sysctl_loader_conf: []
31
32 - name: "loader: [WARNING] does not report changes in check mode because of empty tmp_
↪ loader_vars"
33   ansible.builtin.debug:
34     msg: "[WARNING] Next task does not report changes in check mode because of empty tmp_
↪ loader_vars"
35   when:

```

(continues on next page)

(continued from previous page)

```

36 - ansible_check_mode|bool
37 - fp_loader_warning|bool
38 - fp_loader_debug|bool
39
40 - name: "loader: Add items to fp_sysctl_loader_conf"
41   ansible.builtin.set_fact:
42     fp_sysctl_loader_conf: "{{ fp_sysctl_loader_conf + [item.name ~ '=' ~ item.value, ] }
↳ }}"
43   loop: "{{ fp_sysctl_conf }}"
44   when:
45     - tmp_loader_vars.stdout_lines|default([])|length > 0
46     - item.name in tmp_loader_vars.stdout_lines
47
48 - name: "loader: Debug fp_sysctl_loader_conf"
49   ansible.builtin.debug:
50     var: fp_sysctl_loader_conf|to_yaml
51   when: fp_loader_debug|bool
52
53 - name: "loader: [WARNING] does not report changes in check mode because of empty fp_
↳ sysctl_loader_conf"
54   ansible.builtin.debug:
55     msg: "[WARNING] Next task does not report changes in check mode because of empty fp_
↳ sysctl_loader_conf"
56   when:
57     - ansible_check_mode|bool
58     - fp_loader_warning|bool
59     - fp_loader_debug|bool
60
61 - name: "loader: Configure {{ fp_loader_conf_file }} from template {{ fp_loader_conf_
↳ template }}"
62   ansible.builtin.template:
63     src: "{{ fp_loader_conf_template }}"
64     dest: "{{ fp_loader_conf_file }}"
65     owner: root
66     group: wheel
67     mode: "0644"
68     backup: "{{ fp_backup }}"
69   when:
70     - not ansible_check_mode|bool
71     - fp_loader_conf_template_enable|bool
72   notify: loader message reboot
73
74 - name: "loader: Load modules"
75   ansible.builtin.shell:
76     cmd: >
77       kldstat -q -n {{ item }} || kldload -n {{ item }}
78   loop: "{{ fp_loader_conf_modules }}"
79   changed_when: false
80   when: fp_loader_load_modules|bool
81
82 # EOF
83 ...

```

3.1.21 login.yml

Synopsis: Configure login.

Description of the task.

[tasks/login.yml]

```

1 ---
2
3 - name: "login: Create {{ fp_login_conf_file }}"
4   ansible.builtin.template:
5     src: "{{ fp_login_conf_template }}"
6     dest: "{{ fp_login_conf_file }}"
7     owner: root
8     group: wheel
9     mode: "0644"
10    backup: "{{ fp_backup }}"
11    notify: rebuild login_conf
12    when: fp_login_conf_global|bool
13
14 - name: "login: Create local .login_conf"
15   ansible.builtin.template:
16     src: "{{ fp_login_conf_template_local }}"
17     dest: "/home/{{ item.user }}/.login_conf"
18     owner: "{{ item.user }}"
19     group: "{{ item.group|default(item.user) }}"
20     mode: "0644"
21     backup: "{{ fp_backup }}"
22     loop: "{{ fp_login_conf_local }}"
23     loop_control:
24       label: "{{ item.user }}"
25
26 # EOF
27 ...

```

3.1.22 make.yml

Synopsis: Configure make.

Description of the task.

[tasks/make.yml]

```

1 ---
2
3 - name: "make: Debug (fp_make_debug={{ fp_make_debug }})"
4   vars:
5     msg: |-
6       fp_make_conf_template: {{ fp_make_conf_template }}
7       fp_make_conf_file: {{ fp_make_conf_file }}
8       fp_make_conf_owner: {{ fp_make_conf_owner }}
9       fp_make_conf_group: {{ fp_make_conf_group }}
10      fp_make_conf_mode: {{ fp_make_conf_mode }}

```

(continues on next page)

(continued from previous page)

```

11     fp_make_conf:
12         {{ fp_make_conf|to_nice_yaml|indent(2) }}
13 ansible.builtin.debug:
14     msg: "{{ '{ }'.format(msg) }}"
15 when: fp_make_debug|bool
16 tags: fp_make_debug
17
18 - name: "make: Configure {{ fp_make_conf_file }}"
19 ansible.builtin.template:
20     src: "{{ fp_make_conf_template }}"
21     dest: "{{ fp_make_conf_file }}"
22     owner: "{{ fp_make_conf_owner }}"
23     group: "{{ fp_make_conf_group }}"
24     mode: "{{ fp_make_conf_mode }}"
25     backup: "{{ fp_backup }}"
26 tags: fp_make_conf
27
28 # EOF
29 ...

```

3.1.23 motd.yml

Synopsis: Configure motd.

Description of the task.

[tasks/motd.yml]

```

1 ---
2
3 - name: "motd: Configure /etc/motd"
4 ansible.builtin.template:
5     src: "{{ fp_motd_conf_template }}"
6     dest: "{{ fp_motd_conf_file }}"
7     mode: "{{ fp_motd_conf_mode }}"
8     backup: "{{ fp_backup }}"
9 tags: fp_motd_create
10
11 - name: "motd: Set update_motd YES"
12 ansible.builtin.lineinfile:
13     dest: /etc/rc.conf
14     regexp: '^\\s*update_motd\\s*= (.*)$'
15     line: 'update_motd="YES"'
16     backup: "{{ fp_backup }}"
17 when: fp_update_motd|bool
18 tags: fp_motd_rcconf
19
20 - name: "motd: Set update_motd NO"
21 ansible.builtin.lineinfile:
22     dest: /etc/rc.conf
23     regexp: '^\\s*update_motd\\s*= (.*)$'
24     line: 'update_motd="NO"'

```

(continues on next page)

(continued from previous page)

```

25 backup: "{{ fp_backup }}"
26 when: not fp_update_motd|bool
27 tags: fp_motd_rcconf
28
29 # NOTES
30 # By default system info is added to the top of /etc/motd on reboot.
31 # Set fp_update_motd=False to disable this update.
32
33 # EOF
34 ...

```

3.1.24 mount-img.yml

Synopsis: Configure mount-img.

Description of the task.

[tasks/mount-img.yml]

```

1 ---
2
3 - name: "mount-img: Debug (fp_mount_img_debug={{ fp_mount_img_debug }})"
4   vars:
5     msg: |-
6       fp_mount_img_points_create: {{ fp_mount_img_points_create }}
7       fp_mount_img_entries:
8         {{ fp_mount_img_entries|to_nice_yaml|indent(2) }}
9   ansible.builtin.debug:
10    msg: "{{ '{ }'.format(msg) }}"
11  when: fp_mount_img_debug|bool
12  tags: fp_mount_img_debug
13
14 - name: "mount-img: Set variables"
15   ansible.builtin.set_fact:
16     _fp_mount_task: img
17     _fp_mount_fstype_default: ufs
18     _fp_mount_points_create: "{{ fp_mount_img_points_create }}"
19     _fp_mount_entries: "{{ fp_mount_img_entries }}"
20     _fp_mount_debug: "{{ fp_mount_img_debug }}"
21
22 - name: "mount-iso: Import mount-vnode.yml"
23   ansible.builtin.import_tasks: mount-vnode.yml
24
25 # EOF
26 ...

```

3.1.25 mount-iso.yml

Synopsis: Configure mount-iso.

Description of the task.

[tasks/mount-iso.yml]

```

1 ---
2
3 - name: "mount-iso: Debug (fp_mount_iso_debug={{ fp_mount_iso_debug }})"
4   vars:
5     msg: |-
6       fp_mount_iso_points_create: {{ fp_mount_iso_points_create }}
7       fp_mount_iso_entries:
8         {{ fp_mount_iso_entries|to_nice_yaml|indent(2) }}
9     ansible.builtin.debug:
10      msg: "{{ '{ }'.format(msg) }}"
11    when: fp_mount_iso_debug|bool
12    tags: fp_mount_iso_debug
13
14 - name: "mount-iso: Set variables"
15   ansible.builtin.set_fact:
16     _fp_mount_task: iso
17     _fp_mount_fstype_default: cd9660
18     _fp_mount_points_create: "{{ fp_mount_iso_points_create }}"
19     _fp_mount_entries: "{{ fp_mount_iso_entries }}"
20     _fp_mount_debug: "{{ fp_mount_iso_debug }}"
21
22 - name: "mount-iso: Import mount-vnode.yml"
23   ansible.builtin.import_tasks: mount-vnode.yml
24
25 # EOF
26 ...

```

3.1.26 mount-vnode.yml

Synopsis: Configure mount-vnode.

Description of the task.

[tasks/mount-vnode.yml]

```

1 ---
2
3 - name: "mount-{{ _fp_mount_task }}: Debug (fp_mount_vnode_debug={{ fp_mount_vnode_
4   ↪debug }})"
5   vars:
6     msg: |-
7       _fp_mount_task: {{ _fp_mount_task }}
8       _fp_mount_fstype_default: {{ _fp_mount_fstype_default }}
9       _fp_mount_points_create: {{ _fp_mount_points_create }}
10      _fp_mount_entries:
11        {{ _fp_mount_entries|to_nice_yaml|indent(2) }}

```

(continues on next page)

(continued from previous page)

```

11  ansible.builtin.debug:
12    msg: "{{ '{ }'.format(msg) }}"
13  when: fp_mount_vnode_debug|default(false)|bool
14
15  # mounted - - - - -
16  - name: "mount-{{ _fp_mount_task }}: Create mount points"
17  ansible.builtin.file:
18    state: directory
19    path: "{{ item.mount }}"
20    mode: "{{ item.dirmode|default('0755') }}"
21  loop: "{{ fp_mount_entries }}"
22  when:
23    - _fp_mount_points_create|bool # Note 2.
24    - item.state|default("mounted") == "mounted"
25
26  - name: "mount-{{ _fp_mount_task }}: Attach memory disks"
27  ansible.builtin.include_tasks:
28    file: fn/mdconfig-attach-disk.yml
29  loop: "{{ _fp_mount_entries }}"
30  when: item.state|default("mounted") == "mounted"
31
32  - name: "mount-{{ _fp_mount_task }}: Debug attach memory disks"
33  ansible.builtin.debug:
34    var: al_mdconfig_devs
35  when: _fp_mount_debug|bool
36
37  # TODO: Sanity al_mdconfig_devs[_vnode] is defined; not in check-mode
38
39  - name: "mount-{{ _fp_mount_task }}: Mount"
40  ansible.posix.mount:
41    src: "/dev/md{{ al_mdconfig_devs[_vnode]['unit'] }}{{ item.partition|default('') }}"
42    path: "{{ item.mount }}" # Note 2.
43    fstype: "{{ item.fstype|default(_fp_mount_fstype_default) }}"
44    state: mounted
45    fstab: /tmp/tmp.fstab # Note 1.
46  loop: "{{ _fp_mount_entries }}"
47  register: result
48  when:
49    - al_mdconfig_devs[_vnode] is defined
50    - item.state|default("mounted") == "mounted"
51  vars:
52    _vnode: "{{ item[_fp_mount_task] }}"
53
54  - name: "mount-{{ _fp_mount_task }}: Debug mount"
55  ansible.builtin.debug:
56    var: result
57  when: _fp_mount_debug|bool
58
59  # unmounted - - - - -
60  - name: "mount-{{ _fp_mount_task }}: Unmount"
61  ansible.posix.mount:
62    path: "{{ item.mount }}" # Note 2.

```

(continues on next page)

(continued from previous page)

```

63     state: unmounted
64     fstab: /tmp/tmp.fstab # Note 1.
65     loop: "{{ _fp_mount_entries }}"
66     register: result
67     when: item.state|default("mounted") == "unmounted"
68
69 - name: "mount-{{ _fp_mount_task }}: Debug unmount"
70   ansible.builtin.debug:
71     var: result
72   when: _fp_mount_debug|bool
73
74 - name: "mount-{{ _fp_mount_task }}: Detach memory disks"
75   ansible.builtin.include_tasks:
76     file: fn/mdconfig-detach-disk.yml
77   loop: "{{ _fp_mount_entries }}"
78   when: item.state|default("mounted") == "unmounted"
79
80 - name: "mount-{{ _fp_mount_task }}: Debug detach memory disks"
81   ansible.builtin.debug:
82     var: al_mdconfig_devs
83   when: _fp_mount_debug|bool
84
85 # NOTES
86 # Note 1: It's not possible to mount a device without touching fstab. Temporary fstab is a
87 # workaround.
88 # https://github.com/ansible-collections/ansible.posix/issues/84#issuecomment-742420345
89 # Note 2: Ansible module *mount* will create/remove non-existent mountpoint
90
91 # TODO: Permanent mount entry in fstab
92 # state: present, absent (See ansible.posix.mount parameter *state*)
93 # See /etc/rc.d/mdconfig2
94 # Ex: mdconfig_md0="-t vnode -f /usr/local/data/vdisks/devltd.net"
95 # https://forums.freebsd.org/threads/mounting-vnode-backed-memory-disks-at-startup.21002/
96
97 # EOF
98 ...

```

3.1.27 nfsd.yml

Synopsis: Configure NFS server.

To display variables (3) enable `fp_nfsd_debug=true` (51). Collect status of the services (55) stored in the list `fp_nfsd_service_paths`. The module `service_facts` doesn't work in FreeBSD. Instead, use the task `al_bsd_service_facts` (58) from the library (57). This task doesn't support the check mode. This task will return the dictionary `al_ansible_facts_services`. Display this dictionary (66). This dictionary is used to select the handlers in the dictionary `fp_nfsd_handlers`. Display the dictionary `fp_nfsd_handlers` (72). The defaults are used in check mode because of empty dictionary `al_ansible_facts_services`. Configure `/etc/exports` (78) and notify handler `reload mountd`. Configure and enable (86), or disable (115) services. If the configuration of a service changes the service will be reloaded or restarted when already running.

[tasks/nfsd.yml]

```

1 ---
2
3 - name: "nfsd: Debug (fp_nfsd_debug={{ fp_nfsd_debug }})"
4   vars:
5     msg: |-
6       fp_nfsd_exports:
7         {{ fp_nfsd_exports|to_yaml|indent(2) }}
8       fp_nfsd_enable: {{ fp_nfsd_enable }}
9       fp_nfs_server_flags: {{ fp_nfs_server_flags }}
10      fp_nfs_server_managegids: {{ fp_nfs_server_managegids }}
11      fp_nfs_reserved_port_only: {{ fp_nfs_reserved_port_only }}
12      fp_nfs_bufpackets: {{ fp_nfs_bufpackets }}
13      fp_weak_mountd_authentication: {{ fp_weak_mountd_authentication }}
14      fp_nfsv4_server_enable: {{ fp_nfsv4_server_enable }}
15      fp_nfscbd_enable: {{ fp_nfscbd_enable }}
16      fp_nfscbd_flags: {{ fp_nfscbd_flags }}
17      fp_nfsuserd_enable: {{ fp_nfsuserd_enable }}
18      fp_nfsuserd_flags: {{ fp_nfsuserd_flags }}
19      fp_nfsd_enable_rcconf:
20        {{ fp_nfsd_enable_rcconf|to_yaml|indent(2) }}
21      fp_nfsd_disable_rcconf:
22        {{ fp_nfsd_disable_rcconf|to_yaml|indent(2) }}
23      fp_mountd_enable: {{ fp_mountd_enable }}
24      fp_mountd_flags: {{ fp_mountd_flags }}
25      fp_mountd_enable_rcconf:
26        {{ fp_mountd_enable_rcconf|to_yaml|indent(2) }}
27      fp_mountd_disable_rcconf:
28        {{ fp_mountd_disable_rcconf|to_yaml|indent(2) }}
29      fp_rpcbind_enable: {{ fp_rpcbind_enable }}
30      fp_rpcbind_flags: {{ fp_rpcbind_flags }}
31      fp_rpcbind_enable_rcconf:
32        {{ fp_rpcbind_enable_rcconf|to_yaml|indent(2) }}
33      fp_rpcbind_disable_rcconf:
34        {{ fp_rpcbind_disable_rcconf|to_yaml|indent(2) }}
35      fp_rpc_lockd_enable: {{ fp_rpc_lockd_enable }}
36      fp_rpc_lockd_flags: {{ fp_rpc_lockd_flags }}
37      fp_rpc_lockd_enable_rcconf:
38        {{ fp_rpc_lockd_enable_rcconf|to_yaml|indent(2) }}
39      fp_rpc_lockd_disable_rcconf:
40        {{ fp_rpc_lockd_disable_rcconf|to_yaml|indent(2) }}
41      fp_rpc_statd_enable: {{ fp_rpc_statd_enable }}
42      fp_rpc_statd_flags: {{ fp_rpc_statd_flags }}
43      fp_rpc_statd_enable_rcconf:
44        {{ fp_rpc_statd_enable_rcconf|to_yaml|indent(2) }}
45      fp_rpc_statd_disable_rcconf:
46        {{ fp_rpc_statd_disable_rcconf|to_yaml|indent(2) }}
47      fp_nfsd_service_paths:
48        {{ fp_nfsd_service_paths|to_nice_yaml|indent(2) }}
49   ansible.builtin.debug:
50     msg: "{{ '{ }'.format(msg) }}"
51   when: fp_nfsd_debug|bool
52   tags: fp_nfsd_debug
53

```

(continues on next page)

(continued from previous page)

```

54 # Note: The module service_facts doesn't work in FreeBSD.
55 - name: "nfsd: Service state info"
56   ansible.builtin.include_role:
57     name: vbotka.ansible_lib
58     tasks_from: al_bsd_service_facts
59     apply:
60       tags: fp_nfsd_services
61   tags: always
62   when: not ansible_check_mode
63   vars:
64     al_bsd_service_paths: "{{ fp_nfsd_service_paths }}"
65
66 - name: "nfsd: Debug service state info"
67   ansible.builtin.debug:
68     var: al_ansible_facts_services|default({})
69   when: fp_nfsd_debug|bool
70   tags: fp_nfsd_services
71
72 - name: "nfsd: Debug handlers"
73   ansible.builtin.debug:
74     var: fp_nfsd_handlers
75   when: fp_nfsd_debug|bool
76   tags: fp_nfsd_services
77
78 - name: "nfsd: Configure /etc/exports"
79   ansible.builtin.template:
80     src: exports.j2
81     dest: /etc/exports
82     mode: "0644"
83     backup: "{{ fp_backup }}"
84   notify: reload mountd
85   tags: fp_nfsd_exports
86
87 - name: "nfsd: Enable services"
88   ansible.builtin.include_tasks:
89     file: fn/rcconf-item.yml
90   loop:
91     - label: "nfsd: Enable and start nfs server"
92       conf: "{{ fp_nfsd_enable_rcconf }}"
93       when: "{{ fp_nfsd_enable|bool }}"
94       notify: "{{ fp_nfsd_handlers.nfsd }}"
95     - label: "nfsd: Enable and start mountd"
96       conf: "{{ fp_mountd_enable_rcconf }}"
97       when: "{{ fp_mountd_enable|bool }}"
98       notify: "{{ fp_nfsd_handlers.mountd }}"
99     - label: "nfsd: Enable and start rpcbind"
100       conf: "{{ fp_rpcbind_enable_rcconf }}"
101       when: "{{ fp_rpcbind_enable|bool }}"
102       notify: "{{ fp_nfsd_handlers.rpcbind }}"
103     - label: "nfsd: Enable and start lockd"
104       conf: "{{ fp_rpc_lockd_enable_rcconf }}"
105       when: "{{ fp_rpc_lockd_enable|bool }}"

```

(continues on next page)

(continued from previous page)

```
106     notify: "{{ fp_nfsd_handlers.lockd }}"
107   - label: "nfsd: Enable and start statd"
108     conf: "{{ fp_rpc_statd_enable_rcconf }}"
109     when: "{{ fp_rpc_statd_enable|bool }}"
110     notify: "{{ fp_nfsd_handlers.statd }}"
111   loop_control:
112     loop_var: oi
113     label: "{{ oi.label }}"
114   tags: fp_nfsd_enable
115
116 - name: "nfsd: Disable services"
117   ansible.builtin.include_tasks:
118     file: fn/rcconf-item.yml
119   loop:
120     - label: "nfsd: Disable and stop nfs server"
121       conf: "{{ fp_nfsd_disable_rcconf }}"
122       when: "{{ not fp_nfsd_enable|bool }}"
123       notify: disable and stop nfs server
124     - label: "nfsd: Disable and stop mountd"
125       conf: "{{ fp_mountd_disable_rcconf }}"
126       when: "{{ not fp_mountd_enable|bool }}"
127       notify: disable and stop mountd
128     - label: "nfsd: Disable and stop rpcbind"
129       conf: "{{ fp_rpcbind_disable_rcconf }}"
130       when: "{{ not fp_rpcbind_enable|bool }}"
131       notify: disable and stop rpcbind
132     - label: "nfsd: Disable and stop lockd"
133       conf: "{{ fp_rpc_lockd_disable_rcconf }}"
134       when: "{{ not fp_rpc_lockd_enable|bool }}"
135       notify: disable and stop lockd
136     - label: "nfsd: Disable and stop statd"
137       conf: "{{ fp_rpc_statd_disable_rcconf }}"
138       when: "{{ not fp_rpc_statd_enable|bool }}"
139       notify: disable and stop statd
140   loop_control:
141     loop_var: oi
142     label: "{{ oi.label }}"
143   tags: fp_nfsd_disable
144
145 # EOF
146 ...
```

See also:

- Handler *nfsd.yml*
- Template *exports.j2*
- <TBD>

Note:

- <TBD>

Hint:

- <TBD>

Warning:

- <TBD>

3.1.28 nfs.yml

Synopsis: Configure NFS client.

To display variables (3) enable `fp_nfs_debug=true` (27). Collect status of the services (31) stored in the list `fp_nfs_service_paths`. The module `service_facts` doesn't work in FreeBSD. Instead, use the task `al_bsd_service_facts` (34) from the library (33). This task doesn't support the check mode. This task will return the dictionary `al_ansible_facts_services`. Display this dictionary (42). This dictionary is used to select the handlers in the dictionary `fp_nfs_handlers`. There is no status option of the service `nfscld`. Instead of starting the service, restart it if there are any nfs mountpoints, i.e. the file `/var/db/mounttab` exists. Test the existence of the file `var/db/nounttab` (50) and display the result `fp_var_db_mounttab.stat.exists` (56). This variable will be used to select the handler in the dictionary `fp_nfs_handlers`. Display the dictionary `fp_nfsd_handlers` (62). The defaults are used in check mode because of empty dictionary `al_ansible_facts_services`. Configure and enable (68), or disable (89) services. If the configuration of a service changes the service will be reloaded or restarted when already running.

[tasks/nfs.yml]

```

1 ---
2
3 - name: "nfs: Debug (fp_nfs_debug={{ fp_nfs_debug }})"
4   vars:
5     msg: |-
6       fp_nfs_enable: {{ fp_nfs_enable }}
7       fp_nfs_enable_rcconf:
8         {{ fp_nfs_enable_rcconf|to_yaml|indent(2) }}
9       fp_nfs_disable_rcconf:
10        {{ fp_nfs_disable_rcconf|to_yaml|indent(2) }}
11      fp_rpc_lockd_enable: {{ fp_rpc_lockd_enable }}
12      fp_rpc_lockd_flags: {{ fp_rpc_lockd_flags }}
13      fp_rpc_lockd_enable_rcconf:
14        {{ fp_rpc_lockd_enable_rcconf|to_yaml|indent(2) }}
15      fp_rpc_lockd_disable_rcconf:
16        {{ fp_rpc_lockd_disable_rcconf|to_yaml|indent(2) }}
17      fp_rpc_statd_enable: {{ fp_rpc_statd_enable }}
18      fp_rpc_statd_flags: {{ fp_rpc_statd_flags }}
19      fp_rpc_statd_enable_rcconf:
20        {{ fp_rpc_statd_enable_rcconf|to_yaml|indent(2) }}
21      fp_rpc_statd_disable_rcconf:
22        {{ fp_rpc_statd_disable_rcconf|to_yaml|indent(2) }}
23      fp_nfs_service_paths:
24        {{ fp_nfs_service_paths|to_nice_yaml|indent(2) }}
25  ansible.builtin.debug:

```

(continues on next page)

(continued from previous page)

```

26     msg: "{{ '{ }'.format(msg) }}"
27 when: fp_nfs_debug|bool
28 tags: fp_nfs_debug
29
30 # Note: The module service_facts doesn't work in FreeBSD.
31 - name: "nfs: Service state info"
32   ansible.builtin.include_role:
33     name: vbotka.ansible_lib
34     tasks_from: al_bsd_service_facts
35     apply:
36       tags: fp_nfs_services
37   tags: always
38   when: not ansible_check_mode
39   vars:
40     al_bsd_service_paths: "{{ fp_nfs_service_paths }}"
41
42 - name: "nfs: Debug service state info"
43   ansible.builtin.debug:
44     var: al_ansible_facts_services|default({})
45   when: fp_nfs_debug|bool
46   tags: fp_nfs_services
47
48 # Note: There is no status option of nfsclient. If needed, restart the
49 # service when /var/db/mounttab exists.
50 - name: "nfs: Stat /var/db/mounttab"
51   ansible.builtin.stat:
52     path: /var/db/mounttab
53   register: fp_var_db_mounttab
54   tags: fp_nfs_services
55
56 - name: "nfs: Debug stat /var/db/mounttab"
57   ansible.builtin.debug:
58     var: fp_var_db_mounttab.stat.exists
59   when: fp_nfs_debug|bool
60   tags: fp_nfs_services
61
62 - name: "nfs: Debug handlers"
63   ansible.builtin.debug:
64     var: fp_nfs_handlers
65   when: fp_nfs_debug|bool
66   tags: fp_nfs_services
67
68 - name: "nfs: Enable services"
69   ansible.builtin.include_tasks:
70     file: fn/rcconf-item.yml
71   loop:
72     - label: "nfs: Enable and start nfs client"
73       conf: "{{ fp_nfs_enable_rcconf }}"
74       when: "{{ fp_nfs_enable|bool }}"
75       notify: "{{ fp_nfs_handlers.nfsclient }}"
76     - label: "nfs: Enable and start lockd"
77       conf: "{{ fp_rpc_lockd_enable_rcconf }}"

```

(continues on next page)

(continued from previous page)

```

78     when: "{{ fp_rpc_lockd_enable|bool }}"
79     notify: "{{ fp_nfs_handlers.lockd }}"
80   - label: "nfs: Enable and start statd"
81     conf: "{{ fp_rpc_statd_enable_rcconf }}"
82     when: "{{ fp_rpc_statd_enable|bool }}"
83     notify: "{{ fp_nfs_handlers.statd }}"
84   loop_control:
85     loop_var: oi
86     label: "{{ oi.label }}"
87   tags: fp_nfs_enable
88
89 - name: "nfsd: Disable services"
90   ansible.builtin.include_tasks:
91     file: fn/rcconf-item.yml
92   loop:
93     - label: "nfs: Disable and stop nfs client"
94       conf: "{{ fp_nfs_disable_rcconf }}"
95       when: "{{ not fp_nfs_enable|bool }}"
96       notify: disable and stop nfs client
97     - label: "nfs: Disable and stop lockd"
98       conf: "{{ fp_rpc_lockd_disable_rcconf }}"
99       when: "{{ not fp_rpc_lockd_enable|bool }}"
100      notify: disable and stop lockd
101     - label: "nfs: Disable and stop statd"
102       conf: "{{ fp_rpc_statd_disable_rcconf }}"
103       when: "{{ not fp_rpc_statd_enable|bool }}"
104       notify: disable and stop statd
105   loop_control:
106     loop_var: oi
107     label: "{{ oi.label }}"
108   tags: fp_nfs_disable
109
110 # EOF
111 ...

```

See also:

- Handler *nfs.yml*
- Handler *nfsd.yml*
- <TBD>

Note:

- <TBD>

Hint:

- <TBD>

Warning:

- <TBD>

3.1.29 ntpdate.yml

Synopsis: Configure ntpdate.

Description of the task.

[tasks/ntpdate.yml]

```

1 ---
2
3 - name: "ntpdate: Configure ntpdate in /etc/rc.conf"
4   ansible.builtin.lineinfile:
5     dest: /etc/rc.conf
6     regexp: '^\\s*{{ item.key }}\\s*=(.*)$'
7     line: '{{ item.key }}={{ item.value }}'
8     backup: "{{ fp_backup }}"
9     loop: "{{ fp_ntpdate_rcconf }}"
10
11 # EOF
12 ...

```

3.1.30 ntp.yml

Synopsis: Configure ntp.

Description of the task.

[tasks/ntp.yml]

```

1 ---
2
3 - name: "ntp: Enable and start ntpd"
4   ansible.builtin.lineinfile:
5     dest: /etc/rc.conf
6     regexp: '^\\s*ntp_enable\\s*=(.*)$'
7     line: 'ntp_enable="YES"'
8     backup: "{{ fp_backup }}"
9   when: fp_ntp_enable|bool
10  notify: enable and start ntpd
11  tags: fp_ntp_rcconf
12
13 - name: "ntp: Disable and stop ntpd"
14   ansible.builtin.lineinfile:
15     dest: /etc/rc.conf
16     regexp: '^\\s*ntp_enable\\s*=(.*)$'
17     line: 'ntp_enable="NO"'
18     backup: "{{ fp_backup }}"
19   when: not fp_ntp_enable|bool
20  notify: disable and stop ntpd

```

(continues on next page)

(continued from previous page)

```

21 tags: fp_ntp_rcconf
22
23 - name: "ntp: Create /etc/ntp.conf from template"
24   ansible.builtin.template:
25     src: "{{ fp_ntp_conf_template }}"
26     dest: "{{ fp_ntp_conf_file }}"
27     mode: "{{ fp_ntp_conf_mode }}"
28     backup: "{{ fp_backup }}"
29   notify: restart ntpd
30   tags: fp_ntp_conf
31
32 # EOF
33 ...

```

3.1.31 packages-install.yml

Synopsis: Configure packages-install.

Description of the task.

[tasks/packages-install.yml]

```

1 ---
2
3 - name: "packages-install: Get list of packages for {{ item.list }}"
4   ansible.builtin.set_fact:
5     local_pkg_list: "{{ lookup('vars', 'pkg_dict_' ~ ansible_architecture)|json_
↳ query(query) }}"
6   vars:
7     query: "[?pkglist=='{{ item.list }}'].packages"
8
9 - name: "packages-install: Install packages"
10  community.general.pkgng:
11    name: "{{ local_pkg }}"
12    autoremove: "{{ fp_pkg_autoremove|default(omit) }}"
13    cached: "{{ fp_pkg_cached|default(omit) }}"
14    chroot: "{{ fp_pkg_chroot|default(omit) }}"
15    jail: "{{ fp_pkg_jail|default(omit) }}"
16    pkgstite: "{{ fp_pkg_pkgstite|default(omit) }}"
17    rootdir: "{{ fp_pkg_rootdir|default(omit) }}"
18    state: "{{ fp_pkg_state|default(omit) }}"
19  loop: "{{ local_pkg_list }}"
20  loop_control:
21    loop_var: local_pkg
22    label: "{{ local_pkg }}"
23  register: result
24  retries: "{{ freebsd_install_retries }}"
25  until: result is succeeded
26  delay: "{{ freebsd_install_delay }}"
27  when: freebsd_install_method|lower == "packages"
28
29 - name: "packages-install: Install ports"

```

(continues on next page)

(continued from previous page)

```

30 community.general.portinstall:
31   name: "{{ local_pkg }}"
32   use_packages: "{{ freebsd_use_packages }}"
33   loop: "{{ local_pkg_list }}"
34   loop_control:
35     loop_var: local_pkg
36   register: result
37   retries: "{{ freebsd_install_retries }}"
38   until: result is succeeded
39   delay: "{{ freebsd_install_delay }}"
40   when: freebsd_install_method|lower == "ports"
41
42 # EOF
43 ...

```

3.1.32 packages.yml

Synopsis: Configure packages.

Description of the task.

[tasks/packages.yml]

```

1 ---
2
3 - name: "packages: Debug (fp_packages_debug={{ fp_packages_debug }})"
4   vars:
5     msg: |-
6       freebsd_install_method: {{ freebsd_install_method }}
7       freebsd_use_packages: {{ freebsd_use_packages }}
8       freebsd_install_retries: {{ freebsd_install_retries }}
9       freebsd_install_delay: {{ freebsd_install_delay }}
10      fp_install: {{ fp_install }}
11      fp_pkg_autoremove: {{ fp_pkg_autoremove|default('UNDEFINED') }}
12      fp_pkg_cached: {{ fp_pkg_cached|default('UNDEFINED') }}
13      fp_pkg_chroot: {{ fp_pkg_chroot|default('UNDEFINED') }}
14      fp_pkg_jail: {{ fp_pkg_jail|default('UNDEFINED') }}
15      fp_pkg_pkgsite: {{ fp_pkg_pkgsite|default('UNDEFINED') }}
16      fp_pkg_rootdir: {{ fp_pkg_rootdir|default('UNDEFINED') }}
17      fp_pkg_state: {{ fp_pkg_state|default('UNDEFINED') }}
18      fp_packages:
19        {{ fp_packages|sort(attribute='list')|to_yaml|indent(2) }}
20      fp_links:
21        {{ fp_links|to_yaml|indent(2) }}
22      ansible_architecture: {{ ansible_architecture }}
23      pkg_dict_{{ ansible_architecture }}:
24        {{ lookup('vars', 'pkg_dict_' ~ ansible_architecture)|
25          sort(attribute='pkglist')|to_yaml|indent(2) }}
26   ansible.builtin.debug:
27     msg: "{{ '}'}.format(msg) }}"
28   when: fp_packages_debug|bool
29   tags: fp_packages_debug

```

(continues on next page)

(continued from previous page)

```

30
31 - name: "packages: Install packages"
32   ansible.builtin.include_tasks: packages-install.yml
33   loop: "{{ fp_packages }}"
34   when: item.enabled
35
36 # links
37 - name: "packages: Create symbolic links"
38   ansible.builtin.file:
39     src: "{{ item.target }}"
40     dest: "{{ item.link }}"
41     state: link
42     force: true
43   loop: "{{ fp_links }}"
44   tags: fp_packages_links
45
46 # EOF
47 ...

```

3.1.33 passwords.yml

Synopsis: Configure passwords.

Description of the task.

[tasks/passwords.yml]

```

1 ---
2
3 - name: "passwords: Debug (fp_passwords_debug={{ fp_passwords_debug }})"
4   vars:
5     msg: |-
6       fp_passwords_fail_gracefully: {{ fp_passwords_fail_gracefully }}
7       fp_passwords_update_password: {{ fp_passwords_update_password }}
8       fp_users_conf:
9         {% if fp_passwords_debug_classified|bool %}
10          {{ fp_users_conf|default([])|to_nice_yaml|indent(2) }}
11        {% else %}
12          {% for user in fp_users_conf|default([]) %}
13            - userpass: *****
14              {% for k,v in user.items() %}
15                {% if k not in ['userpass'] %}
16                  {{ k }}: {{ v }}
17              {% endif %}
18            {% endfor %}
19          {% endfor %}
20        {% endif %}
21       fp_passwordstore: {{ fp_passwordstore }}
22       fp_passwordstore_install: {{ fp_passwordstore_install }}
23       fp_passwordstore_debug: {{ fp_passwordstore_debug }}
24       fp_passwordstore_backup: {{ fp_passwordstore_backup }}
25       fp_passwordstore_create: {{ fp_passwordstore_create }}

```

(continues on next page)

(continued from previous page)

```

26     fp_passwordstore_length: {{ fp_passwordstore_length }}
27     fp_passwordstore_nosymbols: {{ fp_passwordstore_nosymbols }}
28     fp_passwordstore_overwrite: {{ fp_passwordstore_overwrite }}
29     fp_passwordstore_passwordstore: {{ fp_passwordstore_passwordstore }}
30     fp_passwordstore_returnall: {{ fp_passwordstore_returnall }}
31     fp_passwordstore_subkey: {{ fp_passwordstore_subkey }}
32     fp_passwordstore_idempotent_password_hash: {{ fp_passwordstore_idempotent_password_
↪hash }}
33     fp_passwordstore_hostname_var: {{ fp_passwordstore_hostname_var }}
34     fp_passwordstore_packages:
35         {{ fp_passwordstore_packages|to_nice_yaml|indent(2) }}
36     ansible.builtin.debug:
37         msg: "{{ '{ }'.format(msg) }}"
38     when: fp_passwords_debug|bool
39     tags: fp_passwords_debug
40
41 - name: "passwords: Passwordstore"
42     block:
43         - name: "passwords: Passwordstore: Install packages"
44           ansible.builtin.include_tasks:
45             file: packages-install.yml
46           loop:
47             - "{{ fp_passwordstore_packages }}"
48             - "{{ fp_gpg_packages }}"
49             - "{{ fp_gpg_packages_extra }}"
50           vars:
51             my_delegate_to_localhost: true
52             run_once: true
53             when: fp_passwordstore_install|bool
54         - name: "passwords: Passwordstore: Retrieve, create, or update userpass"
55           ansible.builtin.include_role:
56             name: vbotka.ansible_lib
57             tasks_from: al_pws_user_host.yml
58           vars:
59             al_pws_debug: "{{ fp_passwordstore_debug }}"
60             al_pws_backup: "{{ fp_passwordstore_backup }}"
61             al_pws_create: "{{ fp_passwordstore_create }}"
62             al_pws_length: "{{ fp_passwordstore_length }}"
63             al_pws_nosymbols: "{{ fp_passwordstore_nosymbols }}"
64             al_pws_overwrite: "{{ fp_passwordstore_overwrite }}"
65             al_pws_passwordstore: "{{ fp_passwordstore_passwordstore }}"
66             al_pws_returnall: "{{ fp_passwordstore_returnall }}"
67             al_pws_subkey: "{{ fp_passwordstore_subkey }}"
68             al_pws_idempotent_password_hash: "{{ fp_passwordstore_idempotent_password_hash }}"
↪
69             al_pws_hostname_var: "{{ fp_passwordstore_hostname_var }}"
70             al_pws_query: "{{ fp_users_conf }}"
71           register: result
72         - name: "passwords: Passwordstore: Create empty list my_passwords"
73           ansible.builtin.set_fact:
74             my_passwords: [] # noqa no-log-password
75         - name: "passwords: Passwordstore: Create my_passwords"

```

(continues on next page)

(continued from previous page)

```

76  ansible.builtin.set_fact:
77      my_passwords: "{{ my_passwords +
78                      [item|dict2items|
79                      rejectattr('key', 'equalto', 'userpass')|
80                      list|items2dict|
81                      combine({'update_password': fp_passwords_update_password})] }}"
82  loop: "{{ al_pws_query_result }}"
83  loop_control:
84      label: "{{ item.name }}"
85  no_log: "{{ not fp_passwords_debug_classified }}"
86  - name: "passwords: Passwordstore: Debug my_passwords"
87      ansible.builtin.debug:
88          var: my_passwords
89      when: fp_passwords_debug_classified|bool
90  - name: "passwords: Passwordstore: Include users"
91      ansible.builtin.include_tasks:
92          file: users.yml
93      vars:
94          fp_users: "{{ my_passwords }}"
95  rescue:
96  - name: "passwords: Passwordstore: Debug fail"
97      ansible.builtin.debug:
98          var: result
99      when: fp_passwords_debug_classified|bool
100  - name: "passwords: Passwordstore: Fail"
101      ansible.builtin.fail:
102          msg: "[ERROR] Passwordstore failed."
103      when: not fp_passwords_fail_gracefully|bool
104  when: fp_passwordstore|bool
105  tags: fp_passwords_passwordstore
106
107  # EOF
108  ...

```

3.1.34 periodic.yml

Synopsis: Configure periodic.

Description of the task.

[tasks/periodic.yml]

```

1  ---
2
3  - name: "periodic: Debug (fp_periodic_debug={{ fp_periodic_debug }})"
4      vars:
5          msg: |-
6              fp_periodic_conf_local_file: {{ fp_periodic_conf_local_file }}
7              fp_periodic_conf_local:
8                  {{ fp_periodic_conf_local|to_yaml|indent(2) }}
9      ansible.builtin.debug:
10         msg: "{{ '{{' }}'.format(msg) }}"

```

(continues on next page)

(continued from previous page)

```

11  when: fp_periodic_debug|bool
12  tags: fp_periodic_debug
13
14  - name: "periodic: Configure {{ fp_periodic_conf_local_file }}"
15    ansible.builtin.lineinfile:
16      create: true
17      mode: "0644"
18      dest: "{{ fp_periodic_conf_local_file }}"
19      regexp: '^\\s*{{ item.key }}\\s*=(.*)$'
20      line: '{{ item.key }}={{ item.value }}'
21      backup: "{{ fp_backup }}"
22    loop: "{{ fp_periodic_conf_local }}"
23    tags: fp_periodic_conf_local
24
25  # NOTES
26  #
27  # 13.10. Monitoring Third Party Security Issues
28  # https://www.freebsd.org/doc/handbook/security-pkg.html
29  #
30  # Not Getting Security Reports
31  # https://forums.freebsd.org/threads/not-getting-security-reports.21841/
32  #
33  # Daily security scripts: best practice for jails?
34  # https://forums.freebsd.org/threads/daily-security-scripts-best-practice-for-jails.
35  ↪ 55868/
36
37  # EOF
  ...

```

3.1.35 procmail.yml

Synopsis: Configure procmail.

Description of the task.

[tasks/procmail.yml]

```

1  ---
2
3  - name: "procmail: Remove forwarders"
4    ansible.builtin.file:
5      state: absent
6      dest: "/home/{{ item.user }}/.forward"
7    loop: "{{ fp_procmail_forwarders }}"
8    when: fp_procmail_forwarders_rebuild|bool
9    tags: fp_procmail_forwarders
10
11 - name: "procmail: Configure forwarders"
12   ansible.builtin.blockinfile:
13     create: true
14     dest: "/home/{{ item.user }}/.forward"
15     owner: "{{ item.user }}"

```

(continues on next page)

(continued from previous page)

```

16  mode: "0644"
17  marker: "# {mark} ANSIBLE MANAGED BLOCK"
18  backup: "{{ fp_backup }}"
19  block: |
20      "{{ item.forward }}"
21  loop: "{{ fp_procmail_forwarders }}"
22  tags: fp_procmail_forwarders
23
24 - name: "procmail: Configure .procmailrc"
25  ansible.builtin.blockinfile:
26  create: true
27  dest: "/home/{{ item.user }}/.procmailrc"
28  owner: "{{ item.user }}"
29  mode: "0644"
30  marker: "# {mark} ANSIBLE MANAGED BLOCK"
31  backup: "{{ fp_backup }}"
32  block: |
33      SHELL={{ fp_procmail_shell }}
34      VERBOSE={{ fp_procmail_verbose }}
35      PATH={{ fp_procmail_path }}
36      MAILDIR={{ fp_procmail_maildir }}
37      DEFAULT=/var/mail/{{ item.user }}
38      PMDIR={{ fp_procmail_pmdir }}
39      LOGFILE=$PMDIR/pm.log
40      LOCKFILE=$PMDIR/.lockfile
41      SENDMAIL=/usr/sbin/sendmail
42  loop: "{{ fp_procmail_forwarders }}"
43  tags: fp_procmail_procmailrc
44
45 - name: "procmail: Create .procmail directory"
46  ansible.builtin.file:
47  state: directory
48  mode: "0755"
49  path: "/home/{{ item.user }}/.procmail"
50  owner: "{{ item.user }}"
51  loop: "{{ fp_procmail_forwarders }}"
52  tags: fp_procmail_dir
53
54 - name: "procmail: Create includerc files"
55  ansible.builtin.template:
56  src: "procmailrc-includerc-{{ item.template }}.j2"
57  dest: "/home/{{ item.user }}/.procmail/{{ item.template }}.rc"
58  owner: "{{ item.user }}"
59  mode: "0644"
60  backup: "{{ fp_backup }}"
61  loop: "{{ fp_procmail_includerc }}"
62  loop_control:
63  label: "{{ item.user }} {{ item.template }}"
64  tags: fp_procmail_includerc
65
66 - name: "procmail: Include rc files in .procmailrc"
67  ansible.builtin.lineinfile:

```

(continues on next page)

(continued from previous page)

```

68 create: true
69 dest: "/home/{{ item.user }}/.procmailrc"
70 owner: "{{ item.user }}"
71 mode: "0644"
72 line: "INCLUDERC=$PMDIR/{{ item.template }}.rc"
73 insertafter: EOF
74 backup: "{{ fp_backup }}"
75 loop: "{{ fp_procmail_includerc }}"
76 loop_control:
77   label: "{{ item.user }} {{ item.template }}"
78 tags: fp_procmail_includerc
79
80 # EOF
81 ...

```

3.1.36 resolvconf.yml

Synopsis: Configure resolvconf.

Description of the task.

[tasks/resolvconf.yml]

```

1 ---
2
3 # resolv.conf
4 - name: "resolvconf: Absent lines in {{ fp_resolv_conf_file }}"
5   ansible.builtin.lineinfile:
6     create: true
7     state: absent
8     dest: "{{ fp_resolv_conf_file }}"
9     owner: "{{ fp_resolv_conf_owner }}"
10    group: "{{ fp_resolv_conf_group }}"
11    mode: "{{ fp_resolv_conf_mode }}"
12    regexp: '^\\s*{{ item }}\\s*$'
13    backup: "{{ fp_backup }}"
14    loop: "{{ fp_resolv_conf_absent }}"
15    tags: fp_resolv_conf_absent
16
17 - name: "resolvconf: Present lines in {{ fp_resolv_conf_file }}"
18   ansible.builtin.lineinfile:
19     create: true
20     dest: "{{ fp_resolv_conf_file }}"
21     owner: "{{ fp_resolv_conf_owner }}"
22     group: "{{ fp_resolv_conf_group }}"
23     mode: "{{ fp_resolv_conf_mode }}"
24     regexp: '^\\s*{{ item.key }}(.*)$'
25     line: '{{ item.key }} {{ item.value }}'
26     backup: "{{ fp_backup }}"
27     loop: "{{ fp_resolv_conf }}"
28     tags: fp_resolv_conf
29

```

(continues on next page)

(continued from previous page)

```

30 # resolvconf.conf
31 - name: "resolvconf: Absent lines in {{ fp_resolvconf_conf_file }}"
32   ansible.builtin.lineinfile:
33     create: true
34     state: absent
35     dest: "{{ fp_resolvconf_conf_file }}"
36     owner: "{{ fp_resolvconf_conf_owner }}"
37     group: "{{ fp_resolvconf_conf_group }}"
38     mode: "{{ fp_resolvconf_conf_mode }}"
39     regexp: '^\\s*{{ item }}\\s*$'
40     backup: "{{ fp_backup }}"
41     loop: "{{ fp_resolvconf_conf_absent }}"
42     notify: resolvconf apply
43     tags: fp_resolvconf_conf_absent
44
45 - name: "resolvconf: Present lines in {{ fp_resolvconf_conf_file }}"
46   ansible.builtin.lineinfile:
47     create: true
48     dest: "{{ fp_resolvconf_conf_file }}"
49     owner: "{{ fp_resolvconf_conf_owner }}"
50     group: "{{ fp_resolvconf_conf_group }}"
51     mode: "{{ fp_resolvconf_conf_mode }}"
52     regexp: '^\\s*{{ item.key }}\\s*=(.*)$'
53     line: '{{ item.key }}={{ item.value }}'
54     backup: "{{ fp_backup }}"
55     loop: "{{ fp_resolvconf_conf }}"
56     notify: resolvconf apply
57     tags: fp_resolvconf_conf
58
59 # EOF
60 ...

```

3.1.37 smartd.yml

Synopsis: Configure smartd.

Description of the task.

[tasks/smartd.yml]

```

1 ---
2
3 - name: "smartd: Enable and start smartd"
4   ansible.builtin.lineinfile:
5     dest: /etc/rc.conf
6     regexp: '^\\s*smartd_enable\\s*=(.*)$'
7     line: 'smartd_enable="YES"'
8     backup: "{{ fp_backup }}"
9   when: fp_smartd_enabled|bool
10  notify: enable and start smartd
11  tags: fp_smartd_rcconf
12

```

(continues on next page)

(continued from previous page)

```

13 - name: "smartd: Disable and stop smartd"
14   ansible.builtin.lineinfile:
15     dest: /etc/rc.conf
16     regexp: '^\\s*smartd_enable\\s*=(.*)$'
17     line: 'smartd_enable="NO"'
18     backup: "{{ fp_backup }}"
19   when: not fp_smartd_enabled|bool
20   notify: disable and stop smartd
21   tags: fp_smartd_rcconf
22
23 - name: "smartd: Configure smartd.conf"
24   ansible.builtin.lineinfile:
25     dest: "{{ fp_smartd_conf_file }}"
26     regexp: '^\\s*DEVICESCAN(.*)$'
27     state: absent
28     backup: "{{ fp_backup }}"
29   notify: reload smartd
30   tags: fp_smartd_conf
31
32 - name: "smartd: Configure devices in {{ fp_smartd_conf_file }}"
33   ansible.builtin.lineinfile:
34     create: true
35     dest: "{{ fp_smartd_conf_file }}"
36     regexp: "{{ item.regexp }}"
37     line: "{{ item.line }}"
38     state: "{{ item.state|default(omit) }}"
39     owner: "{{ fp_smartd_conf_owner }}"
40     group: "{{ fp_smartd_conf_group }}"
41     mode: "{{ fp_smartd_conf_mode }}"
42     backup: "{{ fp_backup }}"
43     loop: "{{ fp_smartd_devices }}"
44     notify: reload smartd
45     tags: fp_smartd_conf
46
47 # EOF
48 ...

```

3.1.38 sanity.yml

Synopsis: Configure sanity.

Description of the task.

[tasks/sanity.yml]

```

1 ---
2
3 - name: "sanity: Debug (fp_sanity_debug={{ fp_sanity_debug }})"
4   ansible.builtin.debug:
5     msg: |-
6       ansible_run_tags:
7         {{ ansible_run_tags|to_yaml }}

```

(continues on next page)

(continued from previous page)

```

 8     ansible_skip_tags:
 9         {{ ansible_skip_tags|to_yaml }}
10     fp_sanity_tags:
11         {{ fp_sanity_tags|to_yaml }}
12     when: fp_sanity_debug|bool
13     tags: fp_sanity_debug
14
15 - name: "sanity: Unknown run tags"
16     ansible.builtin.fail:
17         msg: "[ERROR] Unknown run tag(s): {{ _tags_unknown }}"
18     vars:
19         _tags_unknown: "{{ ansible_run_tags|difference(fp_sanity_tags) }}"
20     when:
21         - fp_sanity_tags_fatal|bool
22         - ansible_run_tags|length > 0
23         - _tags_unknown|length > 0
24
25 - name: "sanity: Unknown skip tags"
26     ansible.builtin.fail:
27         msg: "[ERROR] Unknown skip tag(s): {{ _tags_unknown }}"
28     vars:
29         _tags_unknown: "{{ ansible_skip_tags|difference(fp_sanity_tags) }}"
30     when:
31         - fp_sanity_tags_fatal|bool
32         - ansible_skip_tags|length > 0
33         - _tags_unknown|length > 0
34
35 # EOF
36 ...

```

3.1.39 snmpd.yml

Synopsis: Configure snmpd.

Description of the task.

[tasks/snmpd.yml]

```

1 ---
2
3 - name: "snmpd: Enable and start snmpd"
4     ansible.builtin.lineinfile:
5         dest: /etc/rc.conf
6         regexp: '^s*snmpd_enable\s*=(.*)$'
7         line: 'snmpd_enable="YES"'
8         backup: "{{ fp_backup }}"
9     when: fp_snmpd_enabled|bool
10    notify: enable and start snmpd
11    tags: fp_snmpd_rcconf
12
13 - name: "snmpd: Disable and stop snmpd"
14     ansible.builtin.lineinfile:

```

(continues on next page)

(continued from previous page)

```
15     dest: /etc/rc.conf
16     regexp: '^s*snmpd_enable\s*=(.*)$'
17     line: 'snmpd_enable="NO"'
18     backup: "{{ fp_backup }}"
19     when: not fp_snmpd_enabled|bool
20     notify: disable and stop snmpd
21     tags: fp_snmpd_rcconf
22
23 - name: "snmpd: Configure /etc/rc.conf"
24   ansible.builtin.lineinfile:
25     dest: /etc/rc.conf
26     regexp: '^s*{{ item.key }}\s*=(.*)$'
27     line: '{{ item.key }}="{{ item.value }}"'
28     backup: "{{ fp_backup }}"
29     loop: "{{ fp_snmpd_rcconf }}"
30     notify: restart snmpd
31     tags: fp_snmpd_rcconf
32
33 - name: "snmpd: Copy snmpd.conf.example to snmpd.conf if not exist"
34   ansible.builtin.copy:
35     force: false
36     remote_src: true
37     src: /usr/local/share/snmp/snmpd.conf.example
38     dest: /usr/local/etc/snmpd.conf
39     owner: root
40     group: wheel
41     mode: "0644"
42     notify: restart snmpd
43     tags: fp_snmpd_conf_copy
44
45 - name: "snmpd: Configure /usr/local/etc/snmpd.conf"
46   ansible.builtin.lineinfile:
47     path: /usr/local/etc/snmpd.conf
48     regexp: '^s*{{ item.key }}\s*=(.*)$'
49     line: "{{ item.key }} {{ item.value }}"
50     state: "{{ item.state|default(omit) }}"
51     backup: "{{ fp_backup }}"
52     loop: "{{ fp_snmpd_conf }}"
53     notify: restart snmpd
54     tags: fp_snmpd_conf
55
56 # EOF
57 ...
```

3.1.40 sshd.yml

Synopsis: Configure sshd.

Description of the task.

[tasks/sshd.yml]

```

1 ---
2
3 - name: "sshd: Debug (fp_sshd_debug={{ fp_sshd_debug }})"
4   vars:
5     msg: |-
6       fp_sshd_bin: {{ fp_sshd_bin }}
7       fp_sshd_conf_file: {{ fp_sshd_conf_file }}
8       fp_sshd_conf:
9         {{ fp_sshd_conf|to_yaml|indent(2) }}
10    ansible.builtin.debug:
11      msg: "{{ '{ }'.format(msg) }}"
12    when: fp_sshd_debug|bool
13    tags: fp_sshd_debug
14
15 - name: "sshd: Enable and start sshd"
16   ansible.builtin.lineinfile:
17     dest: /etc/rc.conf
18     regexp: '^s*sshd_enable\s*=(.*)$'
19     line: 'sshd_enable="YES"'
20     backup: "{{ fp_backup }}"
21   notify: enable and start sshd
22   when: fp_sshd_enable|bool
23   tags: fp_sshd_rcconf
24
25 - name: "sshd: Disable and stop sshd"
26   ansible.builtin.lineinfile:
27     dest: /etc/rc.conf
28     regexp: '^s*sshd_enable\s*=(.*)$'
29     line: 'sshd_enable="NO"'
30     backup: "{{ fp_backup }}"
31   notify: disable and stop sshd
32   when: not fp_sshd_enable|bool
33   tags: fp_sshd_rcconf
34
35 - name: "sshd: Configure {{ fp_sshd_conf_file }}"
36   ansible.builtin.lineinfile:
37     dest: "{{ fp_sshd_conf_file }}"
38     regexp: '^s*{{ item.key }}\s+(.*)$'
39     insertbefore: '^s*#\s*{{ item.key }}'
40     line: "{{ item.key }} {{ item.value }}"
41     backup: "{{ fp_backup }}"
42     validate: "{{ fp_sshd_bin }} -t -f %s"
43     loop: "{{ fp_sshd_conf }}"
44   notify: reload sshd
45   tags: fp_sshd_config
46

```

(continues on next page)

(continued from previous page)

```
47 # EOF
48 ...
```

3.1.41 ssh.yml

Synopsis: Configure ssh.

Description of the task.

[tasks/ssh.yml]

```
1 ---
2
3 - name: "ssh: Debug (fp_ssh_debug={{ fp_ssh_debug }})"
4   vars:
5     msg: |-
6       fp_ssh_conf_file: {{ fp_ssh_conf_file }}
7       fp_ssh_conf:
8         {{ fp_ssh_conf|to_yaml|indent(2) }}
9   ansible.builtin.debug:
10    msg: "{{ '{ }'.format(msg) }}"
11  when: fp_ssh_debug|bool
12  tags: fp_ssh_debug
13
14 - name: "ssh: Configure {{ fp_ssh_conf_file }}"
15   ansible.builtin.template:
16    src: "{{ fp_ssh_conf_template }}"
17    dest: "{{ fp_ssh_conf_file }}"
18    owner: "{{ fp_ssh_conf_owner }}"
19    group: "{{ fp_ssh_conf_group }}"
20    mode: "{{ fp_ssh_conf_mode }}"
21    backup: "{{ fp_backup }}"
22  tags: fp_ssh_config
23
24 # EOF
25 ...
```

3.1.42 sudoers.yml

Synopsis: Configure sudoers.

Description of the task.

[tasks/sudoers.yml]

```
1 ---
2
3 - name: "sudoers: Configure {{ fp_sudoers_conf_file }}"
4   ansible.builtin.lineinfile:
5     dest: "{{ fp_sudoers_conf_file }}"
6     regexp: '^\\s*{{ item.key }}\\s+(.*)$'
7     line: "{{ item.key }} {{ item.value }}"
```

(continues on next page)

(continued from previous page)

```

8  state: "{{ item.state|default(omit) }}"
9  insertafter: EOF
10 owner: "{{ fp_sudoers_conf_owner }}"
11 group: "{{ fp_sudoers_conf_group }}"
12 mode: "{{ fp_sudoers_conf_mode }}"
13 backup: "{{ fp_backup }}"
14 loop: "{{ fp_sudoers_conf }}"
15
16 # TODO:
17 # (1) configure /usr/local/etc/sudoers.d
18 # (2) configure /usr/local/etc/sudoers.dist
19
20 # EOF
21 ...

```

3.1.43 swap.yml

Synopsis: Configure swap.

Description of the task.

[tasks/swap.yml]

```

1  ---
2
3  - name: "swap: Debug (fp_swap_debug={{ fp_swap_debug }})"
4    vars:
5      msg: |-
6        fp_swap_enable: {{ fp_swap_enable }}
7        fp_swap_file: {{ fp_swap_file|default("UNDEFINED") }}
8        fp_swap_md: {{ fp_swap_md|default("UNDEFINED") }}
9        fp_swap_size: {{ fp_swap_size|default("UNDEFINED") }}
10       fp_swap_stsize: {{ fp_swap_stsize|default("UNDEFINED") }}
11     ansible.builtin.debug:
12       msg: "{{ '}'}.format(msg) }}"
13     when: fp_swap_debug|bool
14     tags: fp_swap_debug
15
16  - name: "swap: Enable swap and add entry in /etc/fstab"
17     block:
18       - name: "swap: Create swapfile {{ fp_swap_file }}"
19         ansible.builtin.shell:
20           cmd: >
21             sh -c
22             'if [ ! -e {{ fp_swap_file }} ]; then
23               printf "create";
24             fi'
25         register: command_result
26         changed_when: command_result.stdout|default("none") == "create"
27         notify: create swap
28       - name: "swap: Create swap entry in /etc/fstab"
29         ansible.builtin.lineinfile:

```

(continues on next page)

```

30     path: /etc/fstab
31     regexp: '^\\s*{{ fp_swap_md }}\\s+(.*)$'
32     line: "{{ fp_swap_md }} none swap sw,file={{ fp_swap_file }},late 0 0"
33     backup: "{{ fp_backup }}"
34     notify: mount swap
35 # ansible.posix.mount:
36 #   name="none"
37 #   src="{{ fp_swap_md }}"
38 #   fstype="swap"
39 #   opts="sw,file={{ fp_swap_file }},late"
40 #   passno="0"
41 #   dump="0"
42 #   state="present"
43 #   backup yes
44 # Note: mount module fstype swap does not work with FreeBSD as
45 # expected. Instead of adding the src all swap mount points present in
46 # the fstab are replaced with src.
47 - name: "swap: Change swapfile {{ fp_swap_file }}"
48   ansible.builtin.shell:
49     cmd: >
50         sh -c
51         'if [ -e {{ fp_swap_file }} ] &&
52         [ "`stat -f %z {{ fp_swap_file }}`" -ne "{{ fp_swap_stsize }}" ]; then
53             printf "change";
54         fi'
55     register: command_result
56     changed_when: command_result.stdout|default("none") == "change"
57     notify: change and mount swap
58 when: fp_swap_enable|bool
59
60 - name: "swap: Disable swap and remove entry from /etc/fstab"
61   ansible.builtin.lineinfile:
62     path: /etc/fstab
63     regexp: '^\\s*{{ fp_swap_md }}\\s+(.*)$'
64     state: absent
65     backup: "{{ fp_backup }}"
66   notify: umount and delete swap
67   when:
68     - not fp_swap_enable|bool
69     - fp_swap_md is defined
70
71 # EOF
72 ...

```


3.1.44 sysctl.yml

Synopsis: Configure sysctl.

Description of the task.

[tasks/sysctl.yml]

```

1 ---
2
3 - name: "sysctl: Debug (fp_sysctl_debug={{ fp_sysctl_debug }})"
4   vars:
5     msg: |-
6       fp_sysctl_warning_loader_needed: {{ fp_sysctl_warning_loader_needed }}
7       fp_sysctl_conf_file: {{ fp_sysctl_conf_file }}
8       fp_sysctl_ignoreerrors: {{ fp_sysctl_ignoreerrors }}
9       fp_sysctl_conf:
10        {{ fp_sysctl_conf|to_yaml|indent(2) }}
11   ansible.builtin.debug:
12     msg: "{{ '{ }'.format(msg) }}"
13   when: fp_sysctl_debug|bool
14
15 - name: "sysctl: Register sysctl variables that are settable via loader"
16   ansible.builtin.command:
17     cmd: sysctl -aNT
18   register: tmp_loader_vars
19   changed_when: false
20
21 - name: "sysctl: Create empty list fp_sysctl_noloader_conf"
22   ansible.builtin.set_fact:
23     fp_sysctl_noloader_conf: []
24
25 - name: "sysctl: [WARNING] Does not report changes in check mode because of empty tmp_
↳ loader_vars"
26   ansible.builtin.debug:
27     msg: "[WARNING] Next task does not report changes in check mode because of empty tmp_
↳ loader_vars"
28   when:
29     - ansible_check_mode|bool
30     - fp_sysctl_debug|bool
31
32 - name: "sysctl: Add items to fp_sysctl_noloader_conf"
33   ansible.builtin.set_fact:
34     fp_sysctl_noloader_conf: "{{ fp_sysctl_noloader_conf + [item] }}"
35   loop: "{{ fp_sysctl_conf }}"
36   when:
37     - tmp_loader_vars.stdout_lines|default([])|length > 0
38     - item.name not in tmp_loader_vars.stdout_lines
39
40 - name: "sysctl: Debug fp_sysctl_noloader_conf"
41   ansible.builtin.debug:
42     var: fp_sysctl_noloader_conf|to_yaml
43   when: fp_sysctl_debug|bool
44

```

(continues on next page)

(continued from previous page)

```
45 - name: "sysctl: [WARNING] Does not report changes in check mode because of empty fp_
↳sysctl_noloader_conf"
46   ansible.builtin.debug:
47     msg: "[WARNING] Next task does not report changes in check mode because of empty fp_
↳sysctl_noloader_conf"
48   when:
49     - ansible_check_mode|bool
50     - fp_sysctl_debug|bool
51
52 - name: "sysctl: Configure {{ fp_sysctl_conf_file }}"
53   ansible.posix.sysctl:
54     sysctl_file: "{{ fp_sysctl_conf_file }}"
55     name: "{{ item.name }}"
56     value: "{{ item.value }}"
57     state: "{{ item.state|default(omit) }}"
58     reload: "{{ item.reload|default(omit) }}"
59     sysctl_set: "{{ item.set|default(omit) }}"
60     ignoreerrors: "{{ fp_sysctl_ignoreerrors|default(omit) }}"
61     loop: "{{ fp_sysctl_noloader_conf }}"
62
63
64 - name: "sysctl: [WARNING] Tuneables present in fp_sysctl_conf but fp_loader not enabled"
65   ansible.builtin.debug:
66     msg: "[WARNING] Tuneables present in fp_sysctl_conf but fp_loader not enabled"
67   when:
68     - not ansible_check_mode|bool
69     - fp_sysctl_debug|bool
70     - fp_sysctl_warning_loader_needed|bool
71     - fp_sysctl_conf|length > fp_sysctl_noloader_conf|length
72     - not fp_loader|bool
73
74 # NOTES
75 # 1) As of v2.6 module sysctl 'optionally performs a /sbin/sysctl -p'
76 # http://ansible-manual.readthedocs.io/en/latest/sysctl\_module.html
77 # 2) freebsd doesn't support -p, so reload the sysctl service
78 # rc,out,err = self.module.run_command('/etc/rc.d/sysctl reload')
79 # https://github.com/ansible/ansible-modules-core/blob/devel/system/sysctl.py
80 # 3) See loader.yml how tuneables are settable via loader.
81 # https://forums.freebsd.org/threads/reload-loader-conf.72010/
82
83 # EOF
84 ...
```

3.1.45 timezone.yml

Synopsis: Configure timezone.

Description of the task.

[tasks/timezone.yml]

```

1 ---
2
3 - name: "timezone: Debug (fp_timezone_debug={{ fp_timezone_debug }})"
4   ansible.builtin.debug:
5     msg: |-
6       fp_zoneinfo: {{ fp_zoneinfo }}
7   when: fp_timezone_debug|bool
8   tags: fp_timezone_debug
9
10 - name: "timezone: set timezone to {{ fp_zoneinfo }}"
11   community.general.timezone:
12     name: "{{ fp_zoneinfo }}"
13   notify: adjust CMOS
14
15 # EOF
16 ...

```

3.1.46 tmpmfs.yml

Synopsis: Configure tmpmfs.

Description of the task.

[tasks/tmpmfs.yml]

```

1 ---
2
3 - name: "tmpmfs: Debug (fp_tmpmfs_debug={{ fp_tmpmfs_debug }})"
4   vars:
5     msg: |-
6       fp_tmpmfs_rcconf:
7         {{ fp_tmpmfs_rcconf|to_yaml|indent(2) }}
8   ansible.builtin.debug:
9     msg: "{{ ' {}'.format(msg) }}"
10  when: fp_tmpmfs_debug|bool
11  tags: fp_tmpmfs_debug
12
13 - name: "tmpmfs: Configure tmpmfs"
14   ansible.builtin.lineinfile:
15     dest: /etc/rc.conf
16     regexp: '^\\s*{{ item.key }}\\s*=(.*)$'
17     line: '{{ item.key }}={{ item.value }}'
18     backup: "{{ fp_backup }}"
19     loop: "{{ fp_tmpmfs_rcconf }}"
20     tags: fp_tmpmfs_rcconf
21

```

(continues on next page)

(continued from previous page)

```
22 # EOF
23 ...
```

3.1.47 users.yml

Synopsis: Configure users.

Description of the task.

[tasks/users.yml]

```
1 ---
2
3 - name: "users: Debug (fp_users_debug={{ fp_users_debug }})"
4   vars:
5     msg: |-
6       fp_users_debug_classified: {{ fp_users_debug_classified }}
7       fp_users_conf:
8         {{ fp_users_conf|to_nice_yaml|indent(2) }}
9   ansible.builtin.debug:
10    msg: "{{ ' {}'.format(msg) }}"
11  when: fp_users_debug|bool
12  tags: fp_users_debug
13
14 - name: "users: Manage user accounts"
15   ansible.builtin.user:
16     name: "{{ item.name }}"
17     append: "{{ item.append|default(omit) }}"
18     authorization: "{{ item.authorization|default(omit) }}"
19     comment: "{{ item.comment|default(omit) }}"
20     create_home: "{{ item.create_home|default(omit) }}"
21     expires: "{{ item.expires|default(omit) }}"
22     force: "{{ item.force|default(omit) }}"
23     generate_ssh_key: "{{ item.generate_ssh_key|default(omit) }}"
24     group: "{{ item.group|default(omit) }}"
25     groups: "{{ item.groups|default(omit) }}"
26     hidden: "{{ item.hidden|default(omit) }}"
27     home: "{{ item.home|default(omit) }}"
28     local: "{{ item.local|default(omit) }}"
29     login_class: "{{ item.login_class|default(omit) }}"
30     move_home: "{{ item.move_home|default(omit) }}"
31     non_unique: "{{ item.non_unique|default(omit) }}"
32     password: "{{ item.password|default(omit) }}"
33     password_lock: "{{ item.password_lock|default(omit) }}"
34     profile: "{{ item.profile|default(omit) }}"
35     remove: "{{ item.remove|default(omit) }}"
36     role: "{{ item.role|default(omit) }}"
37     seuser: "{{ item.seuser|default(omit) }}"
38     shell: "{{ item.shell|default(omit) }}"
39     skeleton: "{{ item.skeleton|default(omit) }}"
40     ssh_key_bits: "{{ item.ssh_key_bits|default(omit) }}"
41     ssh_key_comment: "{{ item.ssh_key_comment|default(omit) }}"
```

(continues on next page)

(continued from previous page)

```

42  ssh_key_file: "{{ item.ssh_key_file|default(omit) }}"
43  ssh_key_passphrase: "{{ item.ssh_key_passphrase|default(omit) }}"
44  ssh_key_type: "{{ item.ssh_key_type|default(omit) }}"
45  state: "{{ item.state|default(omit) }}"
46  system: "{{ item.system|default(omit) }}"
47  uid: "{{ item.uid|default(omit) }}"
48  update_password: "{{ item.update_password|default(omit) }}"
49  loop: "{{ fp_users_conf }}"
50  loop_control:
51    label: "{{ item.name }}"
52  no_log: "{{ not fp_users_debug_classified }}"
53
54  # EOF
55  ...

```

3.1.48 wpasupplicant.yml

Synopsis: Configure wpasupplicant.

Description of the task.

[tasks/wpasupplicant.yml]

```

1  ---
2
3  - name: "wpasupplicant: Debug (fp_wpasupplicant_debug={{ fp_wpasupplicant_debug }})"
4    vars:
5      msg: |-
6        fp_wpasupplicant_backup: {{ fp_wpasupplicant_backup }}
7        fp_wpasupplicant_program: {{ fp_wpasupplicant_program }}
8        fp_wpasupplicant_flags: {{ fp_wpasupplicant_flags }}
9        fp_wpasupplicant_rcconf:
10         {{ fp_wpasupplicant_rcconf|to_yaml|indent(2) }}
11        fp_wpasupplicant_conf_rc: {{ fp_wpasupplicant_conf_rc }}
12        fp_wpasupplicant_conf_file_link: {{ fp_wpasupplicant_conf_file_link }}
13        fp_wpasupplicant_conf_owner: {{ fp_wpasupplicant_conf_owner }}
14        fp_wpasupplicant_conf_group: {{ fp_wpasupplicant_conf_group }}
15        fp_wpasupplicant_conf_mode: {{ fp_wpasupplicant_conf_mode }}
16        fp_wpasupplicant_conf_dir: {{ fp_wpasupplicant_conf_dir }}
17        fp_wpasupplicant_conf_file: {{ fp_wpasupplicant_conf_file }}
18        fp_wpasupplicant_conf_template: {{ fp_wpasupplicant_conf_template }}
19        fp_wpasupplicant_conf_ctrl_interface: {{ fp_wpasupplicant_conf_ctrl_interface }}
20        fp_wpasupplicant_conf_global:
21         {{ fp_wpasupplicant_conf_global|to_yaml|indent(2) }}
22         {% if fp_wpasupplicant_debug_classified %}
23         fp_wpasupplicant_conf:
24         {{ fp_wpasupplicant_conf|to_yaml|indent(2) }}
25         {% endif %}
26    ansible.builtin.debug:
27      msg: "{{ '{ }'.format(msg) }}"
28    when: fp_wpasupplicant_debug|bool
29    tags: fp_wpasupplicant_debug

```

(continues on next page)

(continued from previous page)

```

30
31 - name: "wpasupplicant: Configure /etc/rc.conf"
32   ansible.builtin.lineinfile:
33     dest: /etc/rc.conf
34     regexp: '^s*{{ item.key }}\s*=(.*)$'
35     line: '{{ item.key }}={{ item.value }}'
36     backup: "{{ fp_wpasupplicant_backup }}"
37     loop: "{{ fp_wpasupplicant_rcconf }}"
38     notify: restart netif
39     tags: fp_wpasupplicant_rcconf
40
41 - name: "wpasupplicant: Configure {{ fp_wpasupplicant_conf_file }}.DEV"
42   ansible.builtin.template:
43     src: "{{ fp_wpasupplicant_conf_template }}"
44     dest: "{{ fp_wpasupplicant_conf_file }}.{{ item.dev }}"
45     owner: "{{ fp_wpasupplicant_conf_owner }}"
46     group: "{{ fp_wpasupplicant_conf_group }}"
47     mode: "{{ fp_wpasupplicant_conf_mode }}"
48     backup: "{{ fp_wpasupplicant_backup }}"
49     register: fp_wpa_supplicant_conf_changes
50     loop: "{{ fp_wpasupplicant_conf }}"
51     loop_control:
52       label: "{{ item.dev }}"
53     no_log: "{{ not fp_wpasupplicant_debug_classified }}"
54     tags: fp_wpasupplicant_conf
55
56 - name: "wpasupplicant: Debug: registered to fp_wpa_supplicant_conf_changes"
57   ansible.builtin.debug:
58     msg: "{{ fp_wpa_supplicant_conf_changes }}"
59     no_log: "{{ not fp_wpasupplicant_debug_classified }}"
60     when: fp_wpasupplicant_debug|bool
61
62 - name: "wpasupplicant: Link {{ fp_wpasupplicant_conf_file }}"
63   ansible.builtin.file:
64     state: link
65     src: "{{ fp_wpasupplicant_conf_file }}.{{ fp_wpasupplicant_conf_file_link }}"
66     dest: "{{ fp_wpasupplicant_conf_file }}"
67     force: true
68     when: fp_wpasupplicant_conf_file_link|length > 0
69     tags: fp_wpasupplicant_conf_file_link
70
71 # Use device specific configuration. Example:
72 # /usr/local/sbin/wpa_supplicant -s -B -i wlan1 -c /etc/wpa_supplicant.conf.wlan1
73 - name: "wpasupplicant: Configure /etc/rc.d/wpa_supplicant"
74   ansible.builtin.lineinfile:
75     dest: /etc/rc.d/wpa_supplicant
76     regexp: '^s*{{ item.key }}\s*=(.*)$'
77     line: '{{ item.key }}={{ item.value }}'
78     backup: "{{ fp_wpasupplicant_backup }}"
79     loop:
80       - {key: "conf_file", value: "${wpa_supplicant_conf_file}.${ifn}"}
81     when: fp_wpasupplicant_conf_rc|bool

```

(continues on next page)

(continued from previous page)

```

82 tags: fp_wpasupplicant_conf_rc
83
84 # EOF
85 ...

```

3.1.49 digitalocean.yml

Synopsis: Configure digitalocean.

Description of the task.

[tasks/cloud-providers/digitalocean.yml]

```

1 ---
2
3 - name: "digitalocean: Debug"
4   vars:
5     msg: |-
6       ansible_distribution_major_version: {{ ansible_distribution_major_version }}
7       fp_do_default_user: {{ fp_do_default_user }}
8       fp_do_default_user_remove: {{ fp_do_default_user_remove }}
9       fp_do_default_user_remove_home: {{ fp_do_default_user_remove_home }}
10      fp_do_rcconf_cloudinit_enable: {{ fp_do_rcconf_cloudinit_enable }}
11      fp_do_rcconf_digitaloceanpre_enable: {{ fp_do_rcconf_digitaloceanpre_enable }}
12      fp_do_rcconf_digitalocean_enable: {{ fp_do_rcconf_digitalocean_enable }}
13      fp_do_rcconf:
14        {{ fp_do_rcconf|to_yaml|indent(2) }}
15   ansible.builtin.debug:
16     msg: "{{ ' {}'.format(msg) }}"
17   when: fp_do_debug|bool
18   tags: fp_do_debug
19
20 - name: "digitalocean: Configure /etc/rc.conf"
21   ansible.builtin.lineinfile:
22     dest: /etc/rc.conf
23     regexp: '^\\s*{{ item.key }}\\s*=(.*)$'
24     line: '{{ item.key }}={{ item.value }}'
25     backup: "{{ fp_backup }}"
26     loop: "{{ fp_do_rcconf }}"
27     tags: fp_do_rcconf
28
29 - name: "digitalocean: Remove default user"
30   ansible.builtin.user:
31     name: "{{ fp_do_default_user }}"
32     state: absent
33     remove: "{{ fp_do_default_user_remove_home }}"
34   when: fp_do_default_user_remove|bool
35   tags: fp_do_default_user_remove
36
37 # DO overwrites hostname from rc.conf
38 - name: "hostname: Configure hostname in /etc/rc.digitalocean.d/droplet.conf"
39   ansible.builtin.lineinfile:

```

(continues on next page)

(continued from previous page)

```

40  dest: /etc/rc.digitalocean.d/droplet.conf
41  regexp: '^s*{{ item.key }}\s*=(.*)$'
42  line: '{{ item.key }}="{{ item.value }}"'
43  backup: "{{ fp_backup }}"
44  loop: "{{ fp_do_legacy_rcconf }}"
45  when: ansible_distribution_major_version|int < 11
46  tags: fp_do_hostname_rcconf
47
48  # EOF
49  ...

```

3.1.50 sync-default.yml

Synopsis: Configure sync-default.

Description of the task.

[tasks/copy/sync-default.yml]

```

1  ---
2  # Sync default
3
4  - name: "copy:sync-default: Debug default directory exists"
5    ansible.builtin.debug:
6      msg: "Directory {{ mypath }} exists: {{ mystat }}"
7    vars:
8      mypath: "{{ fp_copy_dir }}/{{ fp_copy_dir_default }}"
9      mystat: "{{ mypath is exists }}"
10   when: fp_copy_debug|bool
11
12  - name: "copy:sync-default: Synchronize default files {{ fp_copy_dir }}/{{ fp_copy_dir_
13    ↪default }}"
14    ansible.posix.synchronize:
15      src: "{{ mypath_src }}"
16      dest: "{{ item.dest }}"
17      archive: "{{ item.archive|default(fp_synchronize_archive) }}"
18      checksum: "{{ item.checksum|default(fp_synchronize_checksum) }}"
19      compress: "{{ item.compress|default(fp_synchronize_compress) }}"
20      copy_links: "{{ item.copy_links|default(fp_synchronize_copy_links) }}"
21      delete: "{{ item.delete|default(fp_synchronize_delete) }}"
22      dirs: "{{ item.dirs|default(fp_synchronize_dirs) }}"
23      existing_only: "{{ item.existing_only|default(fp_synchronize_existing_only) }}"
24      mode: "{{ item.mode|default(fp_synchronize_mode) }}"
25      group: "{{ item.group|default(omit) }}"
26      owner: "{{ item.owner|default(omit) }}"
27      perms: "{{ item.synchronize_perms|default(omit) }}"
28      times: "{{ item.synchronize_times|default(omit) }}"
29      recursive: "{{ item.recursive|default(omit) }}"
30      verify_host: "{{ item.verify_host|default(fp_synchronize_verify_host) }}"
31    register: result
32    loop: "{{ fp_synchronize_default }}"
33    loop_control:

```

(continues on next page)

(continued from previous page)

```

33     label: "{{ item.dest }}"
34 vars:
35     mypath: "{{ fp_copy_dir }}/{{ fp_copy_dir_default }}/{{ item.src }}"
36 when: mypath is exists
37
38 - name: "copy:sync-optional: Debug synchronize default files result"
39 ansible.builtin.debug:
40     var: result
41 when: fp_copy_debug|bool
42
43 # EOF
44 ...

```

3.1.51 sync-optional.yml

Synopsis: Configure sync-optional.

Description of the task.

[tasks/copy/sync-optional.yml]

```

1 ---
2 # Sync optional
3
4 - name: "copy:sync-optional: Debug optional directory exists"
5   ansible.builtin.debug:
6     msg: "Directory {{ mypath }} exists: {{ mystat }}"
7   vars:
8     mypath: "{{ fp_copy_dir }}/{{ fp_copy_dir_optional }}"
9     mystat: "{{ mypath is exists }}"
10  when: fp_copy_debug|bool
11
12 - name: "copy:sync-optional: Synchronize optional files {{ fp_copy_dir }}/{{ fp_copy_dir_
13   ↪optional }}"
14   ansible.posix.synchronize:
15     src: "{{ mypath }}"
16     dest: "{{ item.dest }}"
17     archive: "{{ item.archive|default(fp_synchronize_archive) }}"
18     checksum: "{{ item.checksum|default(fp_synchronize_checksum) }}"
19     compress: "{{ item.compress|default(fp_synchronize_compress) }}"
20     copy_links: "{{ item.copy_links|default(fp_synchronize_copy_links) }}"
21     delete: "{{ item.delete|default(fp_synchronize_delete) }}"
22     dirs: "{{ item.dirs|default(fp_synchronize_dirs) }}"
23     existing_only: "{{ item.existing_only|default(fp_synchronize_existing_only) }}"
24     mode: "{{ item.mode|default(fp_synchronize_mode) }}"
25     group: "{{ item.group|default(omit) }}"
26     owner: "{{ item.owner|default(omit) }}"
27     perms: "{{ item.synchronize_perms|default(omit) }}"
28     times: "{{ item.synchronize_times|default(omit) }}"
29     recursive: "{{ item.recursive|default(omit) }}"
30     verify_host: "{{ item.verify_host|default(fp_synchronize_verify_host) }}"
31   register: result

```

(continues on next page)

(continued from previous page)

```

31 loop: "{{ fp_synchronize_optional }}"
32 loop_control:
33   label: "{{ item.dest }}"
34 vars:
35   mypath: "{{ fp_copy_dir }}/{{ fp_copy_dir_optional }}/{{ item.src }}"
36 when: mypath is exists
37
38 - name: "copy:sync-optional: Debug synchronize optional files result"
39   ansible.builtin.debug:
40     var: result
41   when: fp_copy_debug|bool
42
43 # EOF
44 ...

```

3.1.52 unarch-default.yml

Synopsis: Configure unarch-default.

Description of the task.

[tasks/copy/unarch-default.yml]

```

1 ---
2 # Unarch default
3
4 - name: "copy:unarch-default: Debug default archive exists"
5   ansible.builtin.debug:
6     msg: "Archive {{ mypath }} exists: {{ mystat }}"
7   vars:
8     mypath: "{{ fp_copy_dir }}/{{ fp_copy_arch_default }}"
9     mystat: "{{ mypath is exists }}"
10  when: fp_copy_debug|bool
11
12 - name: "copy:unarch-default: Extract default archive {{ fp_copy_dir }}/{{ fp_copy_arch_
13 ↪ default }}"
14   ansible.builtin.unarchive:
15     src: "{{ fp_copy_dir }}/{{ fp_copy_arch_default }}"
16     dest: "/"
17     keep_newer: "{{ fp_unarchive_keep_newer }}"
18   vars:
19     mypath: "{{ fp_copy_dir }}/{{ fp_copy_arch_default }}"
20   when: mypath is exists
21
22 # EOF
23 ...

```

3.1.53 unarch-optional.yml

Synopsis: Configure unarch-optional.

Description of the task.

[tasks/copy/unarch-optional.yml]

```

1 ---
2 # Unarch optional
3
4 - name: "copy:unarch-optional: Debug optional archive exists"
5   ansible.builtin.debug:
6     msg: "Archive {{ mypath }} exists: {{ mystat }}"
7   vars:
8     mypath: "{{ fp_copy_dir }}/{{ fp_copy_arch_optional }}"
9     mystat: "{{ mypath is exists }}"
10  when: fp_debug|bool
11
12 - name: "copy:unarch-optional: Extract optional archive {{ fp_copy_dir }}/{{ fp_copy_
13   ↪arch_optional }}"
14   ansible.builtin.unarchive:
15     src: "{{ fp_copy_dir }}/{{ fp_copy_arch_optional }}"
16     dest: "/"
17     keep_newer: "{{ fp_unarchive_keep_newer }}"
18   vars:
19     mypath: "{{ fp_copy_dir }}/{{ fp_copy_arch_optional }}"
20   when: mypath is exists
21 # EOF
22 ...

```

3.1.54 cron-rename-entry.yml

Synopsis: Configure cron-rename-entry.

Description of the task.

[tasks/fn/cron-rename-entry.yml]

```

1 ---
2
3 - name: "cron-rename-entry: Stat {{ fp_cron_dir }}/{{ item.user }}"
4   ansible.builtin.stat:
5     path: "{{ fp_cron_dir }}/{{ item.user }}"
6     register: result
7
8 - name: "cron-rename-entry: Rename present crontab entry"
9   ansible.builtin.replace:
10    path: "{{ fp_cron_dir }}/{{ item.user }}"
11    regexp: '^s*#\s*{{ item.name }}$'
12    replace: "#Ansible: {{ item.name }}"
13    backup: "{{ fp_backup }}"
14    when: result.stat.exists

```

(continues on next page)

(continued from previous page)

```
15
16 # EOF
17 ...
```

3.1.55 mdconfig-attach-disk.yml

Synopsis: Configure mdconfig-attach-disk.

Description of the task.

[tasks/fn/mdconfig-attach-disk.yml]

```
1 ---
2
3 - name: "mdconfig-attach-disk: Attach memory disk"
4   ansible.builtin.include_role:
5     name: vbotka.ansible_lib
6     tasks_from: al_mdconfig.yml
7   vars:
8     al_mdconfig_file: "{{ item[_fp_mount_task] }}"
9
10 # EOF
11 ...
```

3.1.56 mdconfig-detach-disk.yml

Synopsis: Configure mdconfig-detach-disk.

Description of the task.

[tasks/fn/mdconfig-detach-disk.yml]

```
1 ---
2
3 - name: "mdconfig-detach-disk: Detach memory disk"
4   ansible.builtin.include_role:
5     name: vbotka.ansible_lib
6     tasks_from: al_mdconfig.yml
7   vars:
8     al_mdconfig_file: "{{ item[_fp_mount_task] }}"
9     al_mdconfig_state: absent
10
11 # EOF
12 ...
```

3.1.57 rconf-item.yml

Synopsis: Configure rconf-item.

Description of the task.

[tasks/fn/rconf-item.yml]

```

1 ---
2
3 - name: "{{ oi.label }}"
4   ansible.builtin.lineinfile:
5     dest: "{{ oi.dest|default('/etc/rc.conf') }}"
6     regexp: '^\\s*{{ item.key }}\\s*=(*.*)$'
7     line: '{{ item.key }}={{ item.value }}'
8     backup: "{{ fp_backup }}"
9     loop: "{{ oi.conf }}"
10    when: oi.when
11    notify: "{{ oi.notify }}"
12
13 # EOF
14 ...

```

3.2 Handlers

3.2.1 main.yml

Synopsis: Main task.

Import handlers.

[handlers/main.yml]

```

1 ---
2 # handlers for freebsd_postinstall
3
4 - name: Import adjkerntz.yml
5   ansible.builtin.import_tasks: adjkerntz.yml
6 - name: Import apcupsd.yml
7   ansible.builtin.import_tasks: apcupsd.yml
8 - name: Import devfs.yml
9   ansible.builtin.import_tasks: devfs.yml
10 - name: Import dhclient.yml
11   ansible.builtin.import_tasks: dhclient.yml
12 - name: Import fstab.yml
13   ansible.builtin.import_tasks: fstab.yml
14 - name: Import hostapd.yml
15   ansible.builtin.import_tasks: hostapd.yml
16 - name: Import hostname.yml
17   ansible.builtin.import_tasks: hostname.yml
18 - name: Import login.yml
19   ansible.builtin.import_tasks: login.yml
20 - name: Import messages.yml

```

(continues on next page)

(continued from previous page)

```
21  ansible.builtin.import_tasks: messages.yml
22  - name: Import newaliases.yml
23  ansible.builtin.import_tasks: newaliases.yml
24  - name: Import nfsd.yml
25  ansible.builtin.import_tasks: nfsd.yml
26  - name: Import nfs.yml
27  ansible.builtin.import_tasks: nfs.yml
28  - name: Import ntpdate.yml
29  ansible.builtin.import_tasks: ntpdate.yml
30  - name: Import ntpd.yml
31  ansible.builtin.import_tasks: ntpd.yml
32  - name: Import qemu.yml
33  ansible.builtin.import_tasks: qemu.yml
34  - name: Import resolvconf.yml
35  ansible.builtin.import_tasks: resolvconf.yml
36  - name: Import smartd.yml
37  ansible.builtin.import_tasks: smartd.yml
38  - name: Import snmpd.yml
39  ansible.builtin.import_tasks: snmpd.yml
40  - name: Import sshd.yml
41  ansible.builtin.import_tasks: sshd.yml
42  - name: Import swap.yml
43  ansible.builtin.import_tasks: swap.yml
44  - name: Import sysctl.yml
45  ansible.builtin.import_tasks: sysctl.yml
46  - name: Import wpasupplicant.yml
47  ansible.builtin.import_tasks: wpasupplicant.yml
48
49
50 # EOF
51 ...
```

3.2.2 adjkerntz.yml

Synopsis: Handler adjkerntz.

Description of the handler.

[handlers/adjkerntz.yml]

```
1  ---
2  # adjkerntz
3  # update the local time CMOS clock and kernel machdep.adjkerntz
4  # variable when timezone changes occur
5  - name: adjust CMOS
6    ansible.builtin.command:
7      cmd: adjkerntz -a
```

3.2.3 apcupsd.yml

Synopsis: Handler apcupsd.

Description of the handler.

[handlers/apcupsd.yml]

```

1 ---
2 # apcupsd
3 - name: enable and start apcupsd
4   ansible.builtin.service: name="apcupsd" state="started" enabled=true
5 - name: disable and stop apcupsd
6   ansible.builtin.service: name="apcupsd" state="stopped" enabled=false
7 - name: restart apcupsd
8   ansible.builtin.service: name="apcupsd" state="restarted"
9   when: fp_apcupsd_enabled|bool

```

3.2.4 devfs.yml

Synopsis: Handler devfs.

Description of the handler.

[handlers/devfs.yml]

```

1 ---
2 # devfs
3 - name: enable and start devfs
4   ansible.builtin.service: name="devfs" state="started" enabled=true
5   when: not ansible_check_mode
6 - name: disable and stop devfs
7   ansible.builtin.service: name="devfs" state="stopped" enabled=false
8   when: not ansible_check_mode
9 - name: restart devfs
10  ansible.builtin.service: name="devfs" state="restarted"
11  when: fp_devfs_enabled|bool

```

3.2.5 dhclient.yml

Synopsis: Handler dhclient.

Description of the handler.

[handlers/dhclient.yml]

```

1 ---
2 # dhclient
3 - name: restart dhclient
4   ansible.builtin.service:
5     name: dhclient
6     state: restarted
7
8 - name: restart dhclient all

```

(continues on next page)

(continued from previous page)

```

9  ansible.builtin.command:
10     cmd: "/etc/rc.d/dhclient restart {{ item }}"
11     loop: "{{ fp_dhclient_ifn }}"
12
13 - name: restart dhclient blocks
14 ansible.builtin.command:
15     cmd: "/etc/rc.d/dhclient restart {{ item }}"
16     loop: "{{ fp_dhclient_conf_changes.results|
17         selectattr('changed')|
18         map(attribute='item.ifn')|
19         unique }}"

```

3.2.6 fstab.yml

Synopsis: Handler fstab.

Description of the handler.

[handlers/fstab.yml]

```

1  ---
2  # fstab
3  # - name: mount all
4  #   command: mount -a
5  - name: Meta noop
6     ansible.builtin.meta: noop

```

3.2.7 hostapd.yml

Synopsis: Handler hostapd.

Description of the handler.

[handlers/hostapd.yml]

```

1  ---
2  # hostapd
3  - name: enable and start hostapd
4     ansible.builtin.service: name="hostapd" state="started" enabled=true
5  - name: disable and stop hostapd
6     ansible.builtin.service: name="hostapd" state="stopped" enabled=false
7  - name: restart hostapd
8     ansible.builtin.service: name="hostapd" state="restarted"
9     when: fp_hostapd_enabled|bool
10 - name: reload hostapd
11     ansible.builtin.service: name="hostapd" state="reloaded"
12     when: fp_hostapd_enabled|bool

```


3.2.8 hostname.yml

Synopsis: Handler hostname.

Description of the handler.

[handlers/hostname.yml]

```
1 ---
2 # hostname
3 - name: set hostname
4   ansible.builtin.command:
5     cmd: "hostname {{ rc_conf_hostname }}"
```

3.2.9 login.yml

Synopsis: Handler login.

Description of the handler.

[handlers/login.yml]

```
1 ---
2 # login.conf
3 - name: rebuild login_conf
4   ansible.builtin.command:
5     cmd: "/usr/bin/cap_mkdb /etc/login.conf"
```

3.2.10 messages.yml

Synopsis: Handler messages.

Description of the handler.

[handlers/messages.yml]

```
1 ---
2 - name: loader message reboot
3   ansible.builtin.debug:
4     msg: "[MESSAGE] Reboot to activate configuration in /boot/loader.conf"
```

3.2.11 newaliases.yml

Synopsis: Handler newaliases.

Description of the handler.

[handlers/newaliases.yml]

```
1 ---
2 # newaliases
3 - name: newaliases
4   ansible.builtin.command:
5     cmd: /usr/bin/newaliases
```

3.2.12 nfsd.yml

Synopsis: Handler nfsd.

Description of the handler.

[handlers/nfsd.yml]

```
1 ---
2 # nfsd
3 - name: enable and start nfs server
4   ansible.builtin.service: name="nfsd" state="started" enabled=true
5 - name: disable and stop nfs server
6   ansible.builtin.service: name="nfsd" state="stopped" enabled=false
7 - name: restart nfs server
8   ansible.builtin.service: name="nfsd" state="restarted"
9   when: fp_nfsd_enable|bool
10 # mountd
11 - name: enable and start mountd
12   ansible.builtin.service: name="mountd" state="started" enabled=true
13 - name: disable and stop mountd
14   ansible.builtin.service: name="mountd" state="stopped" enabled=false
15 - name: reload mountd
16   ansible.builtin.service: name="mountd" state="reloaded"
17   when: fp_mountd_enable|bool
18 # rpcbind
19 - name: enable and start rpcbind
20   ansible.builtin.service: name="rpcbind" state="started" enabled=true
21 - name: disable and stop rpcbind
22   ansible.builtin.service: name="rpcbind" state="stopped" enabled=false
23 - name: restart rpcbind
24   ansible.builtin.service: name="rpcbind" state="restarted"
25   when: fp_rpcbind_enable|bool
26 # lockd
27 - name: enable and start lockd
28   ansible.builtin.service: name="lockd" state="started" enabled=true
29 - name: disable and stop lockd
30   ansible.builtin.service: name="lockd" state="stopped" enabled=false
31 - name: restart lockd
32   ansible.builtin.service: name="lockd" state="restarted"
33   when: fp_rpc_lockd_enable|bool
34 # statd
35 - name: enable and start statd
36   ansible.builtin.service: name="statd" state="started" enabled=true
37 - name: disable and stop statd
38   ansible.builtin.service: name="statd" state="stopped" enabled=false
39 - name: restart statd
40   ansible.builtin.service: name="statd" state="restarted"
41   when: fp_rpc_statd_enable|bool
```

3.2.13 nfs.yml

Synopsis: Handler nfs.

Description of the handler.

[handlers/nfs.yml]

```
1 ---
2 # nfs
3 - name: enable and start nfs client
4   ansible.builtin.service: name="nfsclient" state="started" enabled=true
5 - name: disable and stop nfs client
6   ansible.builtin.service: name="nfsclient" state="stopped" enabled=false
7 - name: restart nfs client
8   ansible.builtin.service: name="nfsclient" state="restarted"
9   when: fp_nfs_enable|bool
```

3.2.14 ntpdate.yml

Synopsis: Handler ntpdate.

Description of the handler.

[handlers/ntpdate.yml]

```
1 ---
2 # ntpdate
3 - name: enable ntpdate
4   ansible.builtin.service: name="ntpdate" enabled=true
5 - name: disable ntpdate
6   ansible.builtin.service: name="ntpdate" enabled=false
```

3.2.15 ntpd.yml

Synopsis: Handler ntpd.

Description of the handler.

[handlers/ntpd.yml]

```
1 ---
2 # ntpd
3 - name: enable and start ntpd
4   ansible.builtin.service: name="ntpd" state="started" enabled=true
5 - name: disable and stop ntpd
6   ansible.builtin.service: name="ntpd" state="stopped" enabled=false
7 - name: restart ntpd
8   ansible.builtin.service: name="ntpd" state="restarted"
9   when: fp_ntp_enable|bool
```

3.2.16 qemu.yml

Synopsis: Handler qemu.

Description of the handler.

[handlers/qemu.yml]

```

1 ---
2 # qemu
3 - name: enable and start qemu
4   ansible.builtin.service: name="{{ fp_qemu_rcvar }}" state="started" enabled=true
5 - name: disable and stop qemu
6   ansible.builtin.service: name="{{ fp_qemu_rcvar }}" state="stopped" enabled=false
7 - name: restart qemu
8   ansible.builtin.service: name="{{ fp_qemu_rcvar }}" state="restarted"
9   when: fp_qemu_enabled|bool
10 - name: reload qemu
11   ansible.builtin.service: name="{{ fp_qemu_rcvar }}" state="reloaded"
12   when: fp_qemu_enabled|bool

```

3.2.17 resolvconf.yml

Synopsis: Handler resolvconf.

Description of the handler.

[handlers/resolvconf.yml]

```

1 ---
2 # resolvconf
3 - name: resolvconf apply
4   ansible.builtin.command:
5     cmd: resolvconf -u

```

3.2.18 smartd.yml

Synopsis: Handler smartd.

Description of the handler.

[handlers/smartd.yml]

```

1 ---
2 # smartd
3 - name: enable and start smartd
4   ansible.builtin.service: name="smartd" state="started" enabled=true
5 - name: disable and stop smartd
6   ansible.builtin.service: name="smartd" state="stopped" enabled=false
7 - name: restart smartd
8   ansible.builtin.service: name="smartd" state="restarted"
9   when: fp_smartd_enabled|bool
10 - name: reload smartd

```

(continues on next page)

(continued from previous page)

```
11 ansible.builtin.service: name="smartd" state="reloaded"  
12 when: fp_smartd_enabled|bool
```

3.2.19 snmpd.yml

Synopsis: Handler snmpd.

Description of the handler.

[handlers/snmpd.yml]

```
1 ---  
2 # snmpd  
3 - name: enable and start snmpd  
4   ansible.builtin.service: name="snmpd" state="started" enabled=true  
5 - name: disable and stop snmpd  
6   ansible.builtin.service: name="snmpd" state="stopped" enabled=false  
7 - name: restart snmpd  
8   ansible.builtin.service: name="snmpd" state="restarted"  
9   when: fp_snmpd_enabled|bool
```

3.2.20 sshd.yml

Synopsis: Handler sshd.

Description of the handler.

[handlers/sshd.yml]

```
1 ---  
2 # sshd  
3 - name: enable and start sshd  
4   ansible.builtin.service: name="sshd" state="started" enabled=true  
5 - name: disable and stop sshd  
6   ansible.builtin.service: name="sshd" state="stopped" enabled=false  
7 - name: restart sshd  
8   ansible.builtin.service: name="sshd" state="restarted"  
9   when: fp_sshd_enable|bool  
10 - name: reload sshd  
11   ansible.builtin.service: name="sshd" state="reloaded"  
12   when: fp_sshd_enable|bool
```

3.2.21 swap.yml

Synopsis: Handler swap.

Description of the handler.

[handlers/swap.yml]

```

1  ---
2  # swap
3  - name: create swap
4    ansible.builtin.command:
5      cmd: >
6        sh -c
7          'dd if=/dev/zero of={{ fp_swap_file }} bs=1m count={{ fp_swap_size }} &&
8            chmod 0600 {{ fp_swap_file }}'
9
10 - name: mount swap
11   ansible.builtin.command:
12     cmd: sh -c 'swapon -aL'
13   register: command_result
14   failed_when: "'adding' not in command_result.stdout"
15   # NOTE 1: swapon creates /dev/mdXX
16   # NOTE 2: Successful mount returns exit code 1
17   # FreeBSD 11.1. Ansible 2.5 command_result:
18   # RUNNING HANDLER [vbotka.freebsd-postinstall : mount swap] ***
19   # fatal: [aaa.bbb.ccc.ddd]: FAILED! => changed=true
20   # cmd: sh -c 'swapon -aL'
21   # delta: '0:00:00.025491'
22   # end: '2018-07-01 05:47:56.905803'
23   # msg: non-zero return code
24   # rc: 1
25   # start: '2018-07-01 05:47:56.880312'
26   # stderr: "
27   # stderr_lines: []
28   # stdout: 'swapon: adding /dev/md99 as swap device'
29   # stdout_lines: <omitted>
30
31 - name: change and mount swap
32   ansible.builtin.command:
33     cmd: >
34       sh -c
35         'swapoff /dev/{{ fp_swap_md }} &&
36           mdconfig -d -u {{ fp_swap_md }} &&
37           dd if=/dev/zero of={{ fp_swap_file }} bs=1m count={{ fp_swap_size }} &&
38             chmod 0600 {{ fp_swap_file }} &&
39             swapon -aL'
40
41 - name: umount and delete swap
42   ansible.builtin.command:
43     cmd: >
44       sh -c
45         'swapoff /dev/{{ fp_swap_md }} &&
46           mdconfig -d -u {{ fp_swap_md }} &&
47           rm {{ fp_swap_file }}'

```

3.2.22 sysctl.yml

Synopsis: Handler sysctl.

Description of the handler.

[handlers/sysctl.yml]

```

1 ---
2 # sysctl
3 - name: reload sysctl
4   ansible.builtin.service: name="sysctl" state="reloaded"

```

3.2.23 wpasupplicant.yml

Synopsis: Handler wpasupplicant.

Description of the handler.

[handlers/wpasupplicant.yml]

```

1 ---
2 # wpasupplicant
3 - name: restart netif
4   ansible.builtin.service: name="netif" state="restarted"
5
6 - name: reconfigure wpa_supplicant
7   ansible.builtin.shell:
8     cmd: >
9       [ -S {{ fp_wpasupplicant_conf_ctrl_interface }}/{{ item.item.dev }} ] &&
10      wpa_cli -p {{ fp_wpasupplicant_conf_ctrl_interface }} -i {{ item.item.dev }}
11 ↪reconfigure ||
12      exit 0
13   when: item.changed
14   loop: "{{ fp_wpa_supplicant_conf_changes }}"
15 # TODO: Wait for OK result. Error: 'RECONFIGURE' command timed out.

```

3.3 Templates

3.3.1 apcupsd.conf.j2

Synopsis: Template apcupsd.conf.

Description of the template.

[templates/apcupsd.conf.j2]

```

1 ## apcupsd.conf v1.1 ##
2 # {{ ansible_managed }}
3 UPSNAME {{ fp_apcupsd_upsname }}
4 UPSCABLE {{ fp_apcupsd_upscable }}
5 UPSTYPE {{ fp_apcupsd_upstype }}
6 LOCKFILE {{ fp_apcupsd_lockfile }}

```

(continues on next page)

(continued from previous page)

```
7 SCRIPTDIR {{ fp_apcupsd_scriptdir }}
8 PWRFAILDIR {{ fp_apcupsd_pwrfaildir }}
9 NOLOGINDIR {{ fp_apcupsd_nologindir }}
10 ONBATTERYDELAY {{ fp_apcupsd_onbatterydelay }}
11 BATTERYLEVEL {{ fp_apcupsd_batterylevel }}
12 MINUTES {{ fp_apcupsd_minutes }}
13 TIMEOUT {{ fp_apcupsd_timeout }}
14 ANNOY {{ fp_apcupsd_annoy }}
15 ANNOYDELAY {{ fp_apcupsd_annoydelay }}
16 NOLOGON {{ fp_apcupsd_nologon }}
17 KILLDELAY {{ fp_apcupsd_killdelay }}
18 NETSERVER {{ fp_apcupsd_netserver }}
19 NISIP {{ fp_apcupsd_nisip }}
20 NISPORT {{ fp_apcupsd_nisport }}
21 EVENTSFILE {{ fp_apcupsd_eventsfile }}
22 EVENTSFILEMAX {{ fp_apcupsd_eventsfilemax }}
23 UPSCLASS {{ fp_apcupsd_upsclass }}
24 UPSMODE {{ fp_apcupsd_upsmode }}
25 STATTIME {{ fp_apcupsd_stattime }}
26 STATFILE {{ fp_apcupsd_statfile }}
27 LOGSTATS {{ fp_apcupsd_logstats }}
```

3.3.2 exports.j2

Synopsis: Template exports.

Description of the template.

[templates/exports.j2]

```
1 # {{ ansible_managed }}
2 {% for item in fp_nfsd_exports %}
3   {{ item.mount_point }} {{ item.flags }}
4 {% endfor %}
```

3.3.3 groupadd.j2

Synopsis: Template groupadd.

Description of the template.

[templates/groupadd.j2]

```
1 #!/usr/local/bin/bash
2 exec {{ fp_group_add }} "$@"
```


3.3.4 groupdel.j2

Synopsis: Template groupdel.

Description of the template.

[templates/groupdel.j2]

```
1 #!/usr/local/bin/bash
2 exec {{ fp_group_del }} "$@"
```

3.3.5 groupmod.j2

Synopsis: Template groupmod.

Description of the template.

[templates/groupmod.j2]

```
1 #!/usr/local/bin/bash
2 exec {{ fp_group_mod }} "$@"
```

3.3.6 hostapd.conf.j2

Synopsis: Template hostapd.conf.

Description of the template.

[templates/hostapd.conf.j2]

```
1 # {{ ansible_managed }}
2 ssid={{ fp_hostapd_ssid }}
3 interface={{ fp_hostapd_interface }}
4 driver={{ fp_hostapd_driver }}
5 ctrl_interface={{ fp_hostapd_ctrl_interface }}
6 ctrl_interface_group={{ fp_hostapd_ctrl_interface_group }}
7 hw_mode={{ fp_hostapd_hw_mode }}
8 channel={{ fp_hostapd_channel }}
9 beacon_int={{ fp_hostapd_beacon_int }}
10 dtim_period={{ fp_hostapd_dtim_period }}
11 max_num_sta={{ fp_hostapd_max_num_sta }}
12 rts_threshold={{ fp_hostapd_rts_threshold }}
13 fragm_threshold={{ fp_hostapd_fragm_threshold }}
14 auth_algs={{ fp_hostapd_auth_algs }}
15 ignore_broadcast_ssid={{ fp_hostapd_ignore_broadcast_ssid }}
16 #
17 wpa={{ fp_hostapd_wpa }}
18 wpa_pairwise={{ fp_hostapd_wpa_pairwise }}
19 wpa_key_mgmt={{ fp_hostapd_wpa_key_mgmt }}
20 wpa_passphrase={{ fp_hostapd_wpa_passphrase }}
21 #
22 logger_syslog={{ fp_hostapd_logger_syslog }}
23 logger_syslog_level={{ fp_hostapd_logger_syslog_level }}
24 logger_stdout={{ fp_hostapd_logger_stdout }}
```

(continues on next page)

(continued from previous page)

```

25 logger_stdout_level={{ fp_hostapd_logger_stdout_level }}
26 #
27 ##### Integrated EAP server
28 eap_server={{ fp_hostapd_eap_server }}
29 #eap_user_file=/etc/hostapd.eap_user
30 #ca_cert=/etc/hostapd.ca.pem
31 #server_cert=/etc/hostapd.server.pem
32 #private_key=/etc/hostapd.server.prv
33 #private_key_passwd=secret passphrase
34 #check_crl=1
35 #dh_file=/etc/hostapd.dh.pem
36 #eap_sim_db=unix:/tmp/hlr_auc_gw.sock
37 #pac_opaque_encr_key=000102030405060708090a0b0c0d0e0f
38 #eap_fast_a_id=test server
39 #tnc=1

```

3.3.7 hosts.j2

Synopsis: Template hosts.

Description of the template.

[templates/hosts.j2]

```

1 # {{ ansible_managed }}
2 {{ fp_hosts_localhost_ipv6 }}      localhost localhost.{{ fp_domain }}
3 {{ fp_hosts_localhost_ipv4 }}      localhost localhost.{{ fp_domain }}
4
5 {% for item in fp_hosts_conf %}
6 {{ item.ip }}      {{ item.fqdn }} {{ item.hostname|default('') }}
7 {% endfor %}
8
9 # EOF

```

3.3.8 hosts-aliases-list.j2

Synopsis: Template hosts-aliases-list.

Description of the template.

[templates/hosts-aliases-list.j2]

```

1 # {{ ansible_managed }}
2 {{ fp_hosts_localhost_ipv6 }}      localhost localhost.{{ fp_domain }}
3 {{ fp_hosts_localhost_ipv4 }}      localhost localhost.{{ fp_domain }}
4
5 {% for item in fp_hosts_conf %}
6 {{ item.ip }}      {{ item.fqdn }}{% for alias in item.aliases|default([]) %} {{ alias }}{
7 ↪% endfor %}{% if item.hostname|default('')|length > 0 %} {{ item.hostname }}
8 {% endif %}
9 {% endfor %}

```

(continues on next page)

(continued from previous page)

```
9
10 # EOF
```

3.3.9 loader.conf.j2

Synopsis: Template loader.conf.

Description of the template.

[templates/loader.conf.j2]

```
1 # {{ ansible_managed }}
2
3 # fp_loader_conf
4 {% for item in fp_loader_conf %}
5 {{ item }}
6 {% endfor %}
7 {% for item in fp_sysctl_loader_conf %}
8 {{ item }}
9 {% endfor %}
10
11 # fp_loader_conf_modules
12 {% for item in fp_loader_conf_modules %}
13 {{ item }}_load="YES"
14 {% endfor %}
15
16 # EOF
```

3.3.10 login.conf.j2

Synopsis: Template login.conf.

Description of the template.

[templates/login.conf.j2]

```
1 # {{ ansible_managed }}
2
3 # system classes
4 {% for item in fp_login_conf %}
5 {{ item.class }}:\
6 {% for citem in item.conf|flatten %}
7 {{ '\t' }}:{{ citem }}: {% if not loop.last %} \ {% endif %}
8
9 {% endfor %}
10 {% endfor %}
11
12 # extra classes
13 {% for item in fp_login_conf_extra %}
14 {{ item.class }}:\
15 {% for citem in item.conf|flatten %}
16 {{ '\t' }}:{{ citem }}: {% if not loop.last %} \ {% endif %}
```

(continues on next page)

(continued from previous page)

```
17
18 {% endfor %}
19 {% endfor %}
20
21 # EOF
```

3.3.11 login_conf.j2

Synopsis: Template login_conf.

Description of the template.

[templates/login_conf.j2]

```
1 # {{ ansible_managed }}
2
3 {{ item.class }}:\
4 {% for citem in item.conf|flatten %}
5 {{ '\t' }}:{{ citem }}: {% if not loop.last %}\ {% endif %}
6
7 {% endfor %}
8
9 # EOF
```

3.3.12 make.conf.j2

Synopsis: Template make.conf.

Description of the template.

[templates/make.conf.j2]

```
1 # {{ ansible_managed }}
2 {% for item in fp_make_conf %}
3 {{ item }}
4 {% endfor %}
```

3.3.13 motd.j2

Synopsis: Template motd.

Description of the template.

[templates/motd.j2]

```
1 {{ fp_motd_message }}
```

3.3.14 ntp.conf.j2

Synopsis: Template ntp.conf.

Description of the template.

[templates/ntp.conf.j2]

```

1 # {{ ansible_managed }}
2 {% if fp_ntp_config_tinker_panic_enabled %}
3 tinker panic 0
4 {% endif %}
5 {% if fp_ntp_config_driftfile %}
6 driftfile {{ fp_ntp_config_driftfile }}
7 {% endif %}
8 {% for option in fp_ntp_config_options %}
9 {{ option }}
10 {% endfor %}
11 {% for pool in fp_ntp_config_pool %}
12 pool {{ pool }}
13 {% endfor %}
14 {% for server in fp_ntp_config_server %}
15 server {{ server }}
16 {% endfor %}
17 {% for peer in fp_ntp_config_peer %}
18 peer {{ peer }}
19 {% endfor %}
20 {% if fp_ntp_config_statistics %}
21 statistics {{ fp_ntp_config_statistics }}
22 {% endif %}
23 {% for filegen in fp_ntp_config_filegen %}
24 filegen {{ filegen }}
25 {% endfor %}
26 {% for listen in fp_ntp_config_listen %}
27 interface listen {{ listen }}
28 {% endfor %}
29 {% for restrict in fp_ntp_config_restrict %}
30 restrict {{ restrict }}
31 {% endfor %}
32 {% if fp_ntp_config_crypto %}
33 crypto
34 {% endif %}
35 {% if fp_ntp_config_includefile %}
36 includefile {{ fp_ntp_config_includefile }}
37 {% endif %}
38 {% if fp_ntp_config_keys %}
39 keys {{ fp_ntp_config_keys }}
40 {% endif %}
41 {% if fp_ntp_config_trustedkey %}
42 trustedkey {{ fp_ntp_config_trustedkey }}
43 {% endif %}
44 {% if fp_ntp_config_requestkey %}
45 requestkey {{ fp_ntp_config_requestkey }}
46 {% endif %}

```

(continues on next page)

(continued from previous page)

```

47 {% if fp_ntp_config_controlkey %}
48 controlkey {{ fp_ntp_config_controlkey }}
49 {% endif %}
50 {% if fp_ntp_config_broadcast %}
51 broadcast {{ fp_ntp_config_broadcast }}
52 {% endif %}
53 {% if fp_ntp_config_broadcastclient %}
54 broadcastclient
55 {% endif %}
56 {% if fp_ntp_config_multicastclient %}
57 multicastclient {{ fp_ntp_config_multicastclient }}
58 {% endif %}
59 {% if fp_ntp_config_leapfile %}
60 leapfile {{ fp_ntp_config_leapfile }}
61 {% endif %}

```

3.3.15 procmailrc-includerc-capability.j2

Synopsis: Template procmailrc-includerc-capability.

Description of the template.

[templates/procmailrc-includerc-capability.j2]

```

1 # Subject: capability; Capability: Send list of capabilities
2 :0i
3 * !^Subject:.*Re:
4 * !^FROM_DAEMON
5 * ^Subject:.*capability
6 | (/usr/local/bin/formail -r -A "From: {{ item.par1 }}"; \
7   grep -h -r "^# Subject.*Capability" {{ item.par2 }}/*rc ) \
8 | $SENDMAIL -oi -t

```

3.3.16 procmailrc-includerc-duplicates.j2

Synopsis: Template procmailrc-includerc-duplicates.

Description of the template.

[templates/procmailrc-includerc-duplicates.j2]

```

1 # get's rid of duplicates
2 :0 Wh: msgid.lock
3 | formail -D 8192 msgid.cache

```

3.3.17 procmailrc-includerc-get-certificate.j2

Synopsis: Template procmailrc-includerc-get-certificate.

Description of the template.

[templates/procmailrc-includerc-get-certificate.j2]

```

1 # Subject: get certificate; Capability: Send fingerprint of certificate
2 :0i
3 * !^Subject:.*Re:
4 * !^FROM_DAEMON
5 * ^Subject:.*get certificate
6 | (/usr/local/bin/formail -r -A "From: {{ item.par1 }}"; {{ item.par2 }}) | $SENDMAIL -
  ↪oi -t

```

3.3.18 procmailrc-includerc-get-fingerprint.j2

Synopsis: Template procmailrc-includerc-get-fingerprint.

Description of the template.

[templates/procmailrc-includerc-get-fingerprint.j2]

```

1 # Subject: get fingerprint; Capability: Send fingerprint of /etc/ssh/ssh_host_ecdsa_key.
  ↪pub
2 :0i
3 * !^Subject:.*Re:
4 * !^FROM_DAEMON
5 * ^Subject:.*get fingerprint
6 | (/usr/local/bin/formail -r -A "From: {{ item.par1 }}"; \
7  ssh-keygen -l -f {{ item.par2 }}; \
8  ssh-keygen -l -E MD5 -f {{ item.par2 }}) \
9  | $SENDMAIL -oi -t

```

3.3.19 procmailrc-includerc-get-fingerprints.j2

Synopsis: Template procmailrc-includerc-get-fingerprints.

Description of the template.

[templates/procmailrc-includerc-get-fingerprints.j2]

```

1 # Subject: get fingerprints; Capability: Send fingerprints of /etc/ssh/*.pub
2 :0i
3 * !^Subject:.*Re:
4 * !^FROM_DAEMON
5 * ^Subject:.*get fingerprints
6 | (/usr/local/bin/formail -r -A "From: {{ item.par1 }}"; \
7  for i in /etc/ssh/*.pub; \
8  do printf "$i\n"; \
9  ssh-keygen -l -f $i; \
10 ssh-keygen -l -E MD5 -f $i; \

```

(continues on next page)

(continued from previous page)

```
11 printf "\n"; done) \  
12 | $SENDMAIL -oi -t
```

3.3.20 procmailrc-includerc-get-message.j2

Synopsis: Template procmailrc-includerc-get-message.

Description of the template.

[templates/procmailrc-includerc-get-message.j2]

```
1 # Subject: get message; Capability: Send content of {{ item.par2 }}  
2 #  
3 # https://www.freebsd.org/cgi/man.cgi?query=procmail  
4 # If someone sends you a mail with the word `retrieve' in the  
5 # subject, the following will automatically send back the contents  
6 # of info_file to the sender. Like in all recipes where we send  
7 # mail, we watch out for mail loops.  
8 #  
9 # :0  
10 # * !^From +YOUR_USERNAME  
11 # * !^Subject:.*Re:  
12 # * !^FROM_DAEMON  
13 # * ^Subject:.*retrieve  
14 # | (formail -r ; cat info_file) | $SENDMAIL -oi -t  
15 #  
16  
17 :0i  
18 * !^Subject:.*Re:  
19 * !^FROM_DAEMON  
20 * ^Subject:.*get message  
21 | (/usr/local/bin/formail -r -A "From: {{ item.par1 }}"; cat {{ item.par2 }} ) |  
↪$SENDMAIL -oi -t
```

3.3.21 ssh_config.j2

Synopsis: Template ssh_config.

Description of the template.

[templates/ssh_config.j2]

```
1 # {{ ansible_managed }}  
2 {% for item in fp_ssh_conf %}  
3 Host {{ item.host }}  
4 {% for conf in item.conf %}  
5     {{ conf.key }} {{ conf.val }}  
6 {% endfor %}  
7 {% endfor %}  
8  
9 # EOF
```


3.3.22 wpa_supplicant.conf.j2

Synopsis: Template wpa_supplicant.conf.

Description of the template.

[templates/wpa_supplicant.conf.j2]

```
1 # {{ ansible_managed }}
2 {% for gvar in fp_wpasupplicant_conf_global %}
3   {{ gvar.key }}={{ gvar.value }}
4 {% endfor %}
5 {% for net in item.network %}
6   {% for nvar in net.conf %}
7     {% if loop.first %}
8
9     network={
10      {% endif %}
11         {{ nvar.key }}={{ nvar.value }}
12      {% if loop.last %}
13      }
14      {% endif %}
15    {% endfor %}
16  {% endfor %}
17
18 # EOF
```


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`

INDEX

C

check the syntax, 20

D

Dry-run, 22

dry-run, 20

F

fp_*_debug, 15

I

idempotency, 22

idempotent, 20

T

tags, 12