

# Ansible FreeBSD jails

## (ezjail – Jail administration framework)

Vladimir Botka (@vbotka) November 13, 2024

Ansible role vbotka.freebsd\_jail ver.2.6.6

[github.com/vbotka/ansible-freebsd-jail/blob/master/doc/freebsd-ezjail.pdf](https://github.com/vbotka/ansible-freebsd-jail/blob/master/doc/freebsd-ezjail.pdf)

### Abstract

In this article, we describe how to manage FreeBSD Jails with Ansible. Introduction describes how to setup Ansible, configure a project, and create jails using Ansible role freebsd\_jail. This role uses ezjail to manage FreeBSD Jails. The second chapter describes FreeBSD Jail Management by ezjail. A simple Ansible project how to use jails is described in the third chapter. Usage of iocage is not described here.

### Table of Contents

- Introduction
  - Setup
  - Recommended reading
  - FreeBSD controller
  - Configure Ansible
  - Remote host requirements
  - Ansible modules, roles, and collections
  - Ansible role freebsd\_jail
  - Install roles
  - Install dependencies
  - Create project
  - Configure network
  - Configure ZFS
  - Configure firewall
  - Mount ISO
  - Create jails
  - Summary of the project
- Manage jails
  - Archive jails
  - Stop and delete jail
  - Restore and start jail
  - Troubleshooting
- Freebsd jails as Ansible remote hosts
- Ansible module iocage
- Legal notice

- License

## Introduction

FreeBSD is not among the Ansible integrated operating systems, but Ansible community ‘strive to be as BSD-friendly as possible’. Quoting from BSD efforts and contributions:

BSD support is important to us at Ansible. Even though the majority of our contributors use and target Linux we have an active BSD community and strive to be as BSD-friendly as possible.

In details, Ansible core and some collections are testing FreeBSD:

- ansible.posix
- community.crypto
- community.general
- community.libvirt

The collection community.general is important because it includes two FreeBSD-specific modules:

- pkgng Package manager for FreeBSD, and
- portinstall Installing packages from FreeBSD ports system

There are approximately 130 other collections of various quality tested mainly with Linux. You can review them and test the plugins (modules, filters, tests, callbacks, ...) with FreeBSD on your own if you want to use them. You’ll very probably succeed if you can use the required version of Python and install all required dependencies. See for example:

- ansible.utils
- community.mysql
- community.postgresql

Ansible is very flexible. You can create collections, playbooks, roles, and plugins on your own and use any of the FreeBSD jail managers. See Ansible roles tagged jail.

In this article, we describe how to use:

- Ansible role freebsd\_jail that uses ezjail, and
- Ansible module iocage that uses what you think it uses

## Setup

In the following examples, the controller was Ubuntu 23.10 (x86\_64 GNU/Linux) with Ansible 2.16

```
(env) > pip list | grep ansible
```

```

ansible                9.3.0
ansible-compat          4.1.11
ansible-core            2.16.4
ansible-lint            24.2.0

(env) > ansible --version
ansible [core 2.16.4]
...
python version = 3.11.6 (main, Oct  8 2023, 05:06:43) \
                [GCC 13.2.0] (/home/admin/env/bin/python)
jinja version = 3.1.3
libyaml = True

```

and the remote host(s) was FreeBSD 14.0-RELEASE (GENERIC amd64) VM with Python 3.9 on Bhyve (TrueNAS-13.0-U6.1). See the inventory below

```

shell> cat ~/.ansible/hosts

[test]
test_23

[test:vars]
ansible_python_interpreter=/usr/local/bin/python3.9

```

**Note:** Use the Python virtual environment to run ansible-\* commands. The prompt `(env) >` means the Python virtual environment was activated. This is used to run all ansible-\* commands. Other examples may use a standard prompt, for example `shell>`

## Recommended reading

It is expected that the reader has basic knowledge of Ansible. You can skip the rest of this section if you are an advanced Ansible user. If you are new to Ansible, you might want to start with:

- Ansible concepts
- Basic Concepts
- Roles
- Working with playbooks
- How to build your inventory

## FreeBSD controller

You should be fine trying FreeBSD also on the controller. The latest FreeBSD ports and packages provide Ansible 2.15

```

[root@test_23 /]$ pkg info | grep ansible
py39-ansible-8.5.0           Radically simple IT automation
py39-ansible-compat-4.1.2    Ansible compatibility goodies

```

```

py39-ansible-core-2.15.6      Radically simple IT automation
py39-ansible-lint-6.17.1_1    Checks playbooks for sub-optimal \
                               practices and behaviour

```

```

[root@test_23 /]$ ansible --version
ansible [core 2.15.6]
...
python version = 3.9.18 (main, Feb 15 2024, 01:16:25) [Clang 16.0.6 \
    (https://github.com/llvm/llvm-project.git \
    llvmmorg-16.0.6-0-g7cbf1 \
    (/usr/local/bin/python3.9)
jinja version = 3.1.2
libyaml = True

```

The previous versions of the Ansible role `freebsd_jail` were tested with Ansible 2.15 (see Ansible role `freebsd_jail` versions) on FreeBSD remote hosts 12.4 and 13.2 (see Ansible role `freebsd_jail` meta). In the *meta* file, you can also see that the role requires collections `ansible.posix` and `community.general`. In FreeBSD, these collections are installed by default in the directory `/usr/local/lib/python3.9/site-packages/ansible_collections/` from the package `sysutils/ansible` (in this case `py39-ansible-8.5.0`)

```

[root@test_23 /usr/ports]$ cat sysutils/ansible/distinfo
TIMESTAMP = 1698697844
SHA256 (ansible-8.5.0.tar.gz) = 327c509bdaf5cdb2489d85c09d2c107e9432 \
    f9874c8bb5c0702a731160915f2d
SIZE (ansible-8.5.0.tar.gz) = 40712390

```

See the Ansible Community Package Release dependency on the *Ansible Core version*.

## Configure Ansible

By default, most (if not all) packages don't install any Ansible configuration. In this case, the Ansible defaults apply. For example,

```

(env) > ansible-config dump | grep DEFAULT_ROLES_PATH
DEFAULT_ROLES_PATH(default) = ['/home/admin/.ansible/roles', \
    '/usr/local/share/py39-ansible/roles', \
    '/usr/local/etc/ansible/roles']

```

Depending on the directory layout of your project, you might want to change the paths to the modules, roles, and inventory. In addition to this, I recommend to change the callback to *yaml* and enable the *pipelining*. The *yaml* callback makes the output easier to read and *pipelining* speeds up the execution of the tasks on the remote host(s)

```

(env) > cat $HOME/.ansible.cfg

```

```

[defaults]
library = $HOME/.ansible/plugins/modules
inventory = $HOME/.ansible/hosts
roles_path = $HOME/.ansible/roles
stdout_callback = yaml

[connection]
pipelining = true

(env) > ansible-config dump | grep ansible.cfg

ANSIBLE_PIPELINING(/home/admin/.ansible.cfg) = True
CONFIG_FILE() = /home/admin/.ansible.cfg
DEFAULT_HOST_LIST(/home/admin/.ansible.cfg) = ['/home/admin/.ansible/hosts']
DEFAULT_MODULE_PATH(/home/admin/.ansible.cfg) = ['/home/admin/.ansible/plugins/modules']
DEFAULT_ROLES_PATH(/home/admin/.ansible.cfg) = ['/home/admin/.ansible/roles']
DEFAULT_STDOUT_CALLBACK(/home/admin/.ansible.cfg) = yaml

```

See Ansible Configuration Settings.

## Remote host requirements

There are few requirements to manage a remote host from a controller by Ansible:

- Configure the connection
- Set the Python interpreter
- Escalate the privilege

On the remote host, it is a good idea to create a dedicated user that will be used as the `remote_user`. To facilitate this configuration, the Ansible role `ansible.contrib.firstboot` provides the script `firstboot-bsd.sh`. Use it to configure the remote host:

Install the Ansible role `ansible`

```
(env) > ansible-galaxy role install vbotka.ansible
```

Copy your public key to the `remote_user` on the remote host

```
(env) > ssh-copy-id admin@test_23
```

Copy the firstboot script to the remote host

```
(env) > scp $HOME/.ansible/roles/vbotka.ansible/contrib/firstboot/firstboot-bsd.sh \
admin@test_23:~
```

Login to the remote host

```
(env) > ssh admin@test_23
```

Escalate the privilege to root and fit the script to your needs

```
[admin@test_23 ~]$ su -  
Password:  
root@test_23:~ $
```

Run the script on the remote host as root

```
root@test_23:~ $ /home/admin/firstboot-bsd.sh
```

The goal is to connect to the remote host without password and escalate the privilege. Test it, for example

```
(env) > ansible test_23 -b -u admin -m setup | grep ansible_distribution_release  
"ansible_distribution_release": "14.0-RELEASE",
```

#### Notes:

- In production, customize the installation image instead. See [ansible-freebsd-custom-image.rtfid.io](https://ansible-freebsd-custom-image.rtfid.io)
- You can use the Ansible role `ansible` to install Ansible both on FreeBSD and Ubuntu. See [ansible-ansible.rtfid.io](https://ansible-ansible.rtfid.io)

### Ansible modules, roles, and collections

Review all roles to understand what will happen in the system when using them. Let's install and review the Ansible role `freebsd_jail`

```
(env) > ansible-galaxy role install vbotka.freebsd_jail
```

To briefly assess the extent of the Ansible code in the role, the Ansible role `ansible` in `contrib/playbooks` provides the playbook `modules-in-role.yml` that lists modules and collections in a role. See what modules are used in the Ansible role `freebsd_jail`:

```
(env) > ansible-playbook \  
$HOME/.ansible/roles/vbotka.ansible/contrib/playbooks/modules-in-role.yml \  
-e my_role_path=$HOME/.ansible/roles/vbotka.freebsd_jail
```

...

#### List of modules

=====

- `ansible.builtin.command`
- `ansible.builtin.debug`
- `ansible.builtin.fail`
- `ansible.builtin.file`
- `ansible.builtin.import_tasks`
- `ansible.builtin.include_role`
- `ansible.builtin.include_tasks`
- `ansible.builtin.include_vars`
- `ansible.builtin.lineinfile`

- `ansible.builtin.meta`
- `ansible.builtin.set_fact`
- `ansible.builtin.shell`
- `ansible.builtin.stat`
- `ansible.builtin.template`
- `ansible.builtin.unarchive`
- `ansible.posix.synchronize`
- `community.general.pkgng`
- `community.general.portinstall`
- `community.general.zfs_facts`

Review the documentation of the modules and make sure the dependencies are installed. For example, the Ansible module `ansible.posix.synchronize` requires *rsync* to be installed both on the controller and remote host.

## Ansible role `freebsd_jail`

At the moment, this role is tested with jailtype ZFS. See:

- Ansible role `freebsd_jail` documentation
- *ezjail* Jail administration framework

## Install roles

```
(env) > ansible-galaxy role install vbotka.freebsd_jail
(env) > ansible-galaxy role install vbotka.freebsd_postinstall
(env) > ansible-galaxy role install vbotka.ansible_lib
```

The role *freebsd\_postinstall* is used by the role *freebsd\_jail* to mount the ISO image (by default, *bsd\_jail\_mount\_iso=True*). The role *ansible\_lib* provides shared tasks. Optionally, install other roles to configure the network, ZFS, and firewall

```
(env) > ansible-galaxy role install vbotka.freebsd_network
(env) > ansible-galaxy role install vbotka.freebsd_zfs
(env) > ansible-galaxy role install vbotka.freebsd_pf
```

## Install dependencies

In this setup, the remote hosts are always FreeBSD, but the controller can be FreeBSD or theoretically any Linux (tested with Ubuntu). From this perspective, there is no list of required dependences for the roles used in this article. It's up to you to review the tasks in the roles and install the dependences in the OS of your choice. These packages often differ among the brands and sometimes also among the releases of the same brand. For example, the package *jmespath* is required on the controller by the function `community.general.json_query` in *freebsd\_jail* and many other roles. Install it in the Python virtual environment from PyPI

```
(env) > python3 -m pip install jmespath
```

The package *rsync* is required by the Ansible module `ansible.posix.synchronize` both on the controller and the remote host. Install the *rsync* utility from the OS package. For example, on Ubuntu

```
shell> dpkg -l | grep rsync
ii  rsync  3.2.7-1  amd64      fast, versatile, remote (and local) \
                                file-copying tool
```

and on FreeBSD

```
shell> ssh admin@test_23 pkg info | grep rsync
rsync-3.2.7      Network file distribution/synchronization utility
```

Read the documentation of the roles, list and review the used tasks, and install the dependencies on the controller and/or on the remote hosts(s) as required. Test it by running a playbook with the option `--syntax-check`

```
(env) > ansible-playbook --syntax-check playbook.yml
```

and then with the options `-check -diff`

```
(env) > ansible-playbook --check --diff playbook.yml
```

## Create project

Create the directory for the project *test\_23*. For example,

```
shell> pwd
/home/admin/.ansible
shell> mkdir test_23
```

## Configure network

```
[root@test_23 /]# ifconfig -a
em0: flags=1008843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST,LOWER_UP> \
    metric 0 mtu 1500 options=4e504bb<RXCSUM,TXCSUM,VLAN_MTU,\
    VLAN_HWTAGGING,JUMBO_MTU,VLAN_HWCSUM,LRO,VLAN_HWFILTER,\
    VLAN_HWTSO,RXCSUM_IPV6,TXCSUM_IPV6,HWSTATS,MEXTPG>
    ether 00:a0:98:7a:b6:c7
    inet 10.1.0.73 netmask 0xffffffff broadcast 10.1.0.255
    inet 10.1.0.51 netmask 0xffffffff broadcast 10.1.0.51
    inet 10.1.0.52 netmask 0xffffffff broadcast 10.1.0.52
    inet 10.1.0.53 netmask 0xffffffff broadcast 10.1.0.53
    media: Ethernet autoselect (1000baseT <full-duplex>)
    status: active
    nd6 options=29<PERFORMNUD,IFDISABLED,AUTO_LINKLOCAL>
lo0: flags=1008049<UP,LOOPBACK,RUNNING,MULTICAST,LOWER_UP> \
    metric 0 mtu 16384
    options=680003<RXCSUM,TXCSUM,LINKSTATE,RXCSUM_IPV6,TXCSUM_IPV6>
```



```

    inet 127.0.0.1 netmask 0xff000000
    inet6 ::1 prefixlen 128
    inet6 fe80::1%lo0 prefixlen 64 scopeid 0x2
    groups: lo
    nd6 options=21<PERFORMNUD,AUTO_LINKLOCAL>
lo1: flags=1008049<UP,LOOPBACK,RUNNING,MULTICAST,LOWER_UP> \
    metric 0 mtu 16384
    options=680003<RXCSUM,TXCSUM,LINKSTATE,RXCSUM_IPV6,TXCSUM_IPV6>
    inet 127.0.2.1 netmask 0xffffffff
    inet 127.0.2.2 netmask 0xffffffff
    inet 127.0.2.3 netmask 0xffffffff
    inet6 fe80::1%lo1 prefixlen 64 scopeid 0x3
    groups: lo
    nd6 options=21<PERFORMNUD,AUTO_LINKLOCAL>
pflog0: flags=1000141<UP,RUNNING,PROMISC,LOWER_UP> \
    metric 0 mtu 33152
    options=0
    groups: pflog

```

**Note:** See Ansible role `freebsd_network`

## Configure ZFS

Create the playbook

```
shell> cat test_23/freebsd-zfs.yml
```

```

---
- hosts: test_23
  remote_user: admin
  become: true
  roles:
    - vbotka.freebsd_zfs

```

Create variables

```
shell> cat test_23/host_vars/test_23/zfs.yml
```

```

---
fzfs_enable: true
fzfs_manage:
  - name: zroot/jails
    state: present
    extra_zfs_properties:
      compression: 'on'
      mountpoint: /local/jails
fzfs_mountpoints:
  - mountpoint: /local/jails
    owner: root

```

```
group: wheel
mode: '0700'
```

Create the filesystem

```
(env) > ansible-playbook test_23/freebsd-zfs.yml
```

Take a look at the filesystem

```
(env) > ssh admin@test_23 zfs list zroot/jails
NAME                USED  AVAIL  REFER  MOUNTPOINT
zroot/jails         384K   22.8G   384K   /local/jails
```

**Note:** You can configure ZFS manually if you don't want to use this role.

### Configure firewall

```
[root@test_23 /]# cat /etc/pf.conf
```

```
# Ansible managed
# template: default-pf.conf.j2
```

```
# MACROS - - - - -
ext_if = "em0"
localnet = "10.1.0.0/24"
logall = "log"
icmp_types = "{ echoreq, unreachable }"
icmp6_types = "{ echoreq, unreachable }"

# TABLES - - - - -
table <sshabuse> persist
# OPTIONS - - - - -
set skip on lo0
set block-policy return
set loginterface $ext_if
# NORMALIZATION - - - - -
scrub in on $ext_if all fragment reassemble
# QUEUING - - - - -
# TRANSLATION - - - - -
nat on $ext_if from $localnet to any -> ($ext_if)
# FILTERING - - - - -
antispoof for $ext_if
anchor "blacklistd/*" in on $ext_if
anchor "f2b/*"
block $logall all
pass inet proto icmp all icmp-type $icmp_types
pass inet6 proto icmp6 all icmp6-type $icmp6_types
pass from { self, $localnet } to any keep state
```

**Note:** See Ansible role `freebsd_pf`

## Mount ISO image

Create the playbook

```
shell> cat test_23/freebsd-postinstall.yml
```

```
---
- hosts: test_23
  remote_user: admin
  become: true
  roles:
    - vbotka.freebsd_postinstall
```

Copy the ISO image to the remote host `test_23`

```
shell> ssh admin@test_23 ls -la /export/images/FreeBSD-14.0-RELEASE-amd64-dvd1.iso
-r--r--r-- 1 root wheel 4541104128 Mar 13 16:31 \
          /export/images/FreeBSD-14.0-RELEASE-amd64-dvd1.iso
```

Create variables

```
shell> cat test_23/host_vars/test_23/fp-mount-iso.yml
```

```
---
fp_mount_iso: true
fp_mount_iso_entries:
  - iso: /export/images/FreeBSD-14.0-RELEASE-amd64-dvd1.iso
    mount: /export/distro/FreeBSD-14.0-RELEASE-amd64-dvd1.iso
    state: mounted
```

Mount the ISO image

```
(env) > ansible-playbook test_23/freebsd-postinstall.yml -t fp_mount_iso
```

Take a look the mountpoint

```
(env) > ssh admin@test_23 ls -la /export/distro/FreeBSD-14.0-RELEASE-amd64-dvd1.iso
total 101
drwxr-xr-x 19 root wheel 4096 Nov 10 10:29 .
drwxr-xr-x  3 root wheel   3 Mar 13 16:33 ..
-rw-r--r--  2 root wheel 1011 Nov 10 10:29 .cshrc
-rw-r--r--  2 root wheel  495 Nov 10 10:29 .profile
drwxr-xr-x  3 root wheel 2048 Nov 10 10:27 .rr_moved
-r--r--r--  1 root wheel 6109 Nov 10 10:29 COPYRIGHT
drwxr-xr-x  2 root wheel 6144 Nov 10 10:27 bin
drwxr-xr-x 14 root wheel 10240 Nov 10 10:29 boot
dr-xr-xr-x  2 root wheel 2048 Nov 10 10:27 dev
drwxr-xr-x 30 root wheel 14336 Nov 10 10:32 etc
drwxr-xr-x  4 root wheel 12288 Nov 10 10:27 lib
```

```

drwxr-xr-x  3 root wheel  2048 Nov 10 10:27 libexec
drwxr-xr-x  2 root wheel  2048 Nov 10 10:27 media
drwxr-xr-x  2 root wheel  2048 Nov 10 10:27 mnt
drwxr-xr-x  2 root wheel  2048 Nov 10 10:27 net
drwxr-xr-x  4 root wheel  2048 Nov 10 10:32 packages
dr-xr-xr-x  2 root wheel  2048 Nov 10 10:27 proc
drwxr-xr-x  2 root wheel  2048 Nov 10 10:27 rescue
drwxr-xr-x  2 root wheel  2048 Nov 10 10:29 root
drwxr-xr-x  2 root wheel 20480 Nov 10 10:28 sbin
drwxrwxrwt  2 root wheel  2048 Nov 10 10:27 tmp
drwxr-xr-x 14 root wheel  2048 Nov 10 10:29 usr
drwxr-xr-x 24 root wheel  4096 Nov 10 10:27 var

```

**Note:** You can configure the mountpoint manually if you don't want to use this role.

## Create jails

Create the playbook

```
shell> cat test_23/freebsd-jail.yml
```

```

---
- hosts: test_23
  remote_user: admin
  become: true
  roles:
    - vbotka.freebsd_jail

```

Customize the role variables. See defaults *roles/vbotka.freebsd\_jail/defaults/main.yml* and examples in *roles/vbotka.freebsd\_jail/vars/main.yml.sample*

```
shell> cat test_23/host_vars/test_23/jail.yml
```

```

---
bsd_jail_objects_dir: "{{ playbook_dir }}/jails/jail.d"
bsd_jail_objects_dir_extension: yaml

bsd_ezjail_use_zfs: 'YES'
bsd_ezjail_use_zfs_for_jails: 'YES'
bsd_ezjail_jailzfs: zroot/jails
bsd_ezjail_jaildir: /local/jails
bsd_ezjail_archivedir: /export/archive/jails/ezjail_archives
bsd_ezjail_ftphost: "file:///export/distro/FreeBSD-14.0-RELEASE-amd64-dvd1.iso/ \
    usr/freebsd-dist"

bsd_ezjail_conf:
  - 'ezjail_use_zfs="{{ bsd_ezjail_use_zfs }}"'
  - 'ezjail_use_zfs_for_jails="{{ bsd_ezjail_use_zfs_for_jails }}"'
  - 'ezjail_jailzfs="{{ bsd_ezjail_jailzfs }}"'

```

```

- 'ezjail_jaildir="{{ bsd_ezjail_jaildir }}"'
- 'ezjail_archivedir="{{ bsd_ezjail_archivedir }}"'
- 'ezjail_ftphost="{{ bsd_ezjail_ftphost }}"'
bsd_ezjail_flavours:
- flavour: default
  archive: "{{ playbook_dir }}/jails/flavours/default.tar"
- flavour: ansible
  archive: "{{ playbook_dir }}/jails/flavours/ansible.tar"

```

Configure jails. See *roles/vbotka.freebsd\_jail/contrib/jail-objects*

```

shell> tree test_23/jails/jail.d/
test_23/jails/jail.d/
--- test_01.yml
--- test_02.yml
--- test_03.yml
--- test_04.yml

```

For example, the jail *test\_01*

```

shell> cat test_23/jails/jail.d/test_01.yml

```

```

---
objects:
- jailname: test_01
  present: true
  start: true
  jailtype: zfs
  flavour: ansible
  interface:
    - {dev: lo1, ip4: 127.0.2.1}
    - {dev: em0, ip4: 10.1.0.51}
  parameters:
    - {key: allow.raw_sockets, val: 'true'}
    - {key: allow.set_hostname, val: 'true'}
  jail_conf:
    - {key: mount.devfs}
  ezjail_conf: []
  archive: test_01-202311060342.38.tar.gz
  firstboot: /root/firstboot.sh
  firstboot_owner: root
  firstboot_group: wheel
  firstboot_mode: '0750'

```

Create flavours. Customize the flavours to your needs. See *roles/vbotka.freebsd\_jail/contrib/jail-flavours*

```

shell> tar tvf test_23/jails/flavours/default.tar
-rw-r--r-- root/wheel      39 2023-11-04 14:52 etc/resolv.conf

```

```

-rwxr-xr-x root/wheel      1821 2023-11-04 14:52 etc/rc.d/ezjail.flavour.default
shell> tar tvf test_23/jails/flavours/ansible.tar
-rw----- admin/admin      1475 2023-11-04 14:52 home/admin/.ssh/authorized_keys
-rwxr-x--- root/wheel       855 2023-11-04 14:52 root/firstboot.sh
-r--r----- root/wheel     3978 2023-11-04 14:54 usr/local/etc/sudoers
-rw-r--r-- root/wheel       39 2023-11-04 14:52 etc/resolv.conf
-rw-r--r-- root/wheel       39 2023-11-04 14:52 etc/rc.conf
-rwxr-xr-x root/wheel      1821 2023-11-04 14:52 etc/rc.d/ezjail.flavour.default

```

Run the playbook and display debug. Take a look at the variables

```

(env) > ansible-playbook test_23/freebsd-jail.yml -t bsd_jail_debug \
                                           -e bsd_jail_debug=true

```

```

TASK [vbotka.freebsd_jail : FreeBSD Jail Debug] *****
ok: [test_23] =>

```

```

msg: |-
  ansible_architecture: amd64
  ansible_os_family: FreeBSD
  ansible_distribution: FreeBSD
  ansible_distribution_major_version: 14
  ansible_distribution_version: 14.0
  ansible_distribution_release: 14.0-RELEASE
  ansible_python_version: 3.9.18
  ansible_interfaces: [em0, lo0, lo1, pflog0]

```

```

  bsd_jail_conf_backup: True
  bsd_jail_install: True
  freebsd_install_method: packages
  freebsd_use_packages: True
  freebsd_install_retries: 10
  freebsd_install_delay: 5
  bsd_jail_packages:
    - sysutils/ezjail

```

```

  bsd_jail_packages_extra:
    []

```

```

  bsd_jail_mount_iso: True
  bsd_jail_assert: True
  bsd_jail_assert_enable:
    interfaces: true
    jaildir: true
    zfs: true

```

```

  bsd_jail: True
  bsd_jail_enable: True

```

```

bsd_jail_service: jail
bsd_jail_conf_path: /etc
bsd_jail_conf_file: /etc/jail.conf
bsd_jail_conf_owner: root
bsd_jail_conf_group: wheel
bsd_jail_conf_mode: 0644
bsd_jail_conf:
    []

bsd_jail_id_dir: /var/run
bsd_jail_stamp_dir: /var/db/jail-stamps
bsd_jail_fstab_dir: /etc/jail
bsd_jail_mount_iso: True
bsd_jail_jails_defaults:
    - {key: path, val: /local/jails/$name}
    - {key: mount.fstab, val: '/etc/jail/fstab.${name}'}
    - {key: exec.start, val: /bin/sh /etc/rc}
    - {key: exec.stop, val: /bin/sh /etc/rc.shutdown}
    - {key: devfs_ruleset, val: '4'}
    - {key: exec.clean}
    - {key: mount.devfs}
    - {key: mount.fdescfs}
    - {key: mount.procfs}

bsd_jail_confd: False
bsd_jail_confd_dir: /etc/jail.conf.d

bsd_jail_objects_dir: /home/admin/.ansible/test_23/jails/jail.d
bsd_jail_objects_dir_extension: yml
bsd_jail_jails_present_names:
    [test_02, test_01, test_03]

bsd_jail_jails_absent_names:
    [test_04]

bsd_jail_jails_present:
    - archive: test_01-202311060342.38.tar.gz
      ezjail_conf: []
      firstboot: /root/firstboot.sh
      firstboot_group: wheel
      firstboot_mode: '0750'
      firstboot_owner: root
      flavour: ansible
      interface:
        - {dev: lo1, ip4: 127.0.2.1}
        - {dev: em0, ip4: 10.1.0.51}

```

```

jail_conf:
- {key: mount.devfs}
jailname: test_01
jailtype: zfs
parameters:
- {key: allow.raw_sockets, val: 'true'}
- {key: allow.set_hostname, val: 'true'}
present: true
start: true
- archive: test_02-202311060342.18.tar.gz
  ezjail_conf: []
  firstboot_group: wheel
  firstboot_mode: '0750'
  firstboot_owner: root
  flavour: ansible
  interface:
  - {dev: lo1, ip4: 127.0.2.2}
  - {dev: em0, ip4: 10.1.0.52}
  jail_conf:
  - {key: mount.devfs}
  jailname: test_02
  jailtype: zfs
  parameters:
  - {key: allow.raw_sockets, val: 'true'}
  - {key: allow.set_hostname, val: 'true'}
  present: true
  start: true
- archive: test_03-202311060341.58.tar.gz
  ezjail_conf: []
  firstboot: /root/firstboot.sh
  firstboot_group: wheel
  firstboot_mode: '0750'
  firstboot_owner: root
  flavour: ansible
  interface:
  - {dev: lo1, ip4: 127.0.2.3}
  - {dev: em0, ip4: 10.1.0.53}
  jail_conf:
  - {key: mount.devfs}
  jailname: test_03
  jailtype: zfs
  parameters:
  - {key: allow.raw_sockets, val: 'true'}
  - {key: allow.set_hostname, val: 'true'}
  present: true
  start: true

```



```

bsd_jail_jails_absent:
  - interface:
    - {dev: lo1, ip4: 127.0.2.4}
    - {dev: em0, ip4: 10.1.0.54}
    jailname: test_04
    jailtype: zfs
    present: false

bsd_ezjail: True
bsd_ezjail_enable: True
bsd_ezjail_service: ezjail
bsd_ezjail_conf_path: /usr/local/etc
bsd_ezjail_conf_file: /usr/local/etc/ezjail.conf
bsd_ezjail_conf_owner: root
bsd_ezjail_conf_group: wheel
bsd_ezjail_conf_mode: 0644
bsd_ezjail_conf:
  - ezjail_use_zfs="YES"
  - ezjail_use_zfs_for_jails="YES"
  - ezjail_jailzfs="zroot/jails"
  - ezjail_jaildir="/local/jails"
  - ezjail_archivedir="/export/archive/jails/ezjail_archives"
  - ezjail_ftphost="file:///export/distro/FreeBSD-14.0-RELEASE-amd64-dvd1.iso/ \
    usr/freebsd-dist"

bsd_ezjail_use_zfs: YES
bsd_ezjail_jailzfs: zroot/jails
bsd_ezjail_jaildir: /local/jails
bsd_ezjail_flavours:
  - archive: /home/admin/.ansible/test_23/jails/flavours/default.tar
    flavour: default
  - archive: /home/admin/.ansible/test_23/jails/flavours/ansible.tar
    flavour: ansible

bsd_ezjail_install_command: install
bsd_ezjail_install_options:
bsd_ezjail_install_force: False
bsd_ezjail_admin_restore: False
bsd_ezjail_admin_restore_options:
bsd_ezjail_restart_jails:
  []

bsd_jail_start: True
bsd_jail_firstboot: True
bsd_jail_shutdown: True

```

```
bsd_jail_info: False
```

Install packages

```
(env) > ansible-playbook test_23/freebsd-jail.yml -t bsd_jail_packages
```

Create flavours

```
(env) > ansible-playbook test_23/freebsd-jail.yml -t bsd_jail_ezjail_flavours
```

Create jails

```
(env) > ansible-playbook test_23/freebsd-jail.yml
```

Login to the remote host and list the jails

```
shell> ssh admin@test_23
```

```
[admin@test_23 ~]$ ezjail-admin list
```

STA	JID	IP	Hostname	Root Directory
ZR	3	127.0.2.3	test_03	/local/jails/test_03
	3	em0 10.1.0.53		
ZR	2	127.0.2.2	test_02	/local/jails/test_02
	2	em0 10.1.0.52		
ZR	1	127.0.2.1	test_01	/local/jails/test_01
	1	em0 10.1.0.51		

### Summary of the project

```
shell> tree -S -L 2 -a .ansible/
```

```
.ansible/
```

```
--- cp
```

```
--- galaxy_cache
```

```
--- galaxy_token
```

```
--- hosts
```

```
--- roles
```

```
- --- vbotka.ansible
```

```
- --- vbotka.ansible_lib
```

```
- --- vbotka.freebsd_jail
```

```
- --- vbotka.freebsd_network
```

```
- --- vbotka.freebsd_pf
```

```
- --- vbotka.freebsd_postinstall
```

```
- --- vbotka.freebsd_zfs
```

```
--- test_23
```

```
- --- freebsd-jail.yml
```

```
- --- freebsd-pf.yml
```

```
- --- freebsd-postinstall.yml
```

```
- --- freebsd-zfs.yml
```

```
- --- host_vars
```

```

-   --- jails
--- tmp

shell> tree -S -a .ansible/test_23/host_vars/
.ansible/test_23/host_vars/
--- test_23
    --- fp-mount-iso.yml
    --- jail.yml
    --- pf.yml
    --- zfs.yml

shell> tree -S -a .ansible/test_23/jails/
.ansible/test_23/jails/
--- flavours
-   --- ansible
-   -   --- etc
-   -   -   rc.conf
-   -   -   rc.d
-   -   -   ezjail.flavour.default
-   -   -   resolv.conf
-   -   --- home
-   -   -   admin
-   -   -   .ssh
-   -   -   authorized_keys
-   -   --- root
-   -   -   firstboot.sh
-   -   --- usr
-   -   -   local
-   -   -   etc
-   -   -   sudoers
-   --- ansible-list.txt
-   --- ansible.tar
-   --- ansible.tar.orig
-   --- default-list.txt
-   --- default.tar
-   --- README
--- jail.d
    --- test_01.yml
    --- test_02.yml
    --- test_03.yml
    --- test_04.yml

```

The playbooks and the roles are idempotent

```

(env) > export ANSIBLE_DISPLAY_OK_HOSTS=false
(env) > export ANSIBLE_DISPLAY_SKIPPED_HOSTS=false
(env) > ansible-playbook test_23/freebsd-zfs.yml

```

```
PLAY [test_23] *****
```

```
TASK [Include vbotka.freebsd_postinstall sysctl] *****
```

```
TASK [Include vbotka.freebsd_postinstall loader] *****
```

```
TASK [vbotka.freebsd_zfs : facts: Get list of datasets] *****
included: /home/admin/.ansible/roles/vbotka.freebsd_zfs/tasks/facts_include_1.yml \
  for test_23 => (item=['zroot', 'zroot/ROOT', 'zroot/ROOT/default', \
    'zroot/home', 'zroot/jails', 'zroot/jails/basejail', \
    'zroot/jails/basejail@20240318_21:27:35', 'zroot/jails/newjail', \
    'zroot/jails/test_01', 'zroot/jails/test_02', 'zroot/jails/test_03', \
    'zroot/tmp', 'zroot/usr', 'zroot/usr/ports', 'zroot/usr/src', \
    'zroot/var', 'zroot/var/audit', 'zroot/var/crash', 'zroot/var/log', \
    'zroot/var/mail', 'zroot/var/tmp'])
```

```
PLAY RECAP *****
```

```
test_23: ok=15 changed=0 unreachable=0 failed=0 skipped=21 rescued=0 ignored=0
```

```
(env) > ansible-playbook test_23/freebsd-pf.yml
```

```
PLAY [test_23] *****
```

```
PLAY RECAP *****
```

```
test_23: ok=22 changed=0 unreachable=0 failed=0 skipped=23 rescued=0 ignored=0
```

```
(env) > ansible-playbook test_23/freebsd-jail.yml
```

```
PLAY [test_23] *****
```

```
TASK [Include vbotka.freebsd_postinstall mount-iso] *****
```

```
TASK [vbotka.freebsd_postinstall : mount-iso: Attach memory disks] *****
```

```
included: /home/admin/.ansible/roles/vbotka.freebsd_postinstall/tasks/fn/mdconfig-attach-disk-iso.yml \
  for test_23 => (item={'iso': '/export/images/FreeBSD-14.0-RELEASE-amd64-dvd1.iso', \
    'mount': '/export/distro/FreeBSD-14.0-RELEASE-amd64-dvd1.iso', 'state': 'mounted'})
```

```
TASK [mdconfig-attach-disk: Attach memory disk] *****
```

```
TASK [vbotka.freebsd_jail : ezjail-jails: Delete jails] *****
```

```
included: /home/admin/.ansible/roles/vbotka.freebsd_jail/tasks/fn/delete.yml \
  for test_23 => (item=test_04)
```

```
TASK [vbotka.freebsd_jail : ezjail-jails: Create and configure jails] **
```

```
included: /home/admin/.ansible/roles/vbotka.freebsd_jail/tasks/fn/create.yml \
  for test_23 => (item=test_01)
```

```

included: /home/admin/.ansible/roles/vbotka.freebsd_jail/tasks/fn/create.yml \
  for test_23 => (item=test_02)
included: /home/admin/.ansible/roles/vbotka.freebsd_jail/tasks/fn/create.yml \
  for test_23 => (item=test_03)

TASK [vbotka.freebsd_jail : firstboot: Exec firstboot scripts in the jails] **
included: /home/admin/.ansible/roles/vbotka.freebsd_jail/tasks/fn/firstboot.yml \
  for test_23 => (item=test_01)
included: /home/admin/.ansible/roles/vbotka.freebsd_jail/tasks/fn/firstboot.yml \
  for test_23 => (item=test_02)
included: /home/admin/.ansible/roles/vbotka.freebsd_jail/tasks/fn/firstboot.yml \
  for test_23 => (item=test_03)

PLAY RECAP *****
test_23: ok=63 changed=0 unreachable=0 failed=0 skipped=94 rescued=0 ignored=0

```

## Manage jails

### Archive jails

Shutdown and stop the jails, archive them, and start again

```

[root@test_23 /home/admin]# jexec 1 /etc/rc.shutdown
Stopping sshd.
Waiting for PIDS: 26702.
Stopping cron.
Waiting for PIDS: 26706.
.
Terminated

```

```

[root@test_23 /home/admin]# jexec 2 /etc/rc.shutdown
Stopping sshd.
Waiting for PIDS: 26933.
Stopping cron.
Waiting for PIDS: 26937.
.
Terminated

```

```

[root@test_23 /home/admin]# jexec 3 /etc/rc.shutdown
Stopping sshd.
Waiting for PIDS: 27164.
Stopping cron.
Waiting for PIDS: 27168.
.
Terminated

```

```

[root@test_23 /home/admin]# ezjail-admin stop test_01

```

```
Stopping jails:/etc/rc.d/jail: WARNING: /var/run/jail.test_01.conf is created and used \
for jail test_01.

test_01.
```

```
[root@test_23 /home/admin]# ezjail-admin stop test_02
Stopping jails:/etc/rc.d/jail: WARNING: /var/run/jail.test_02.conf is created and used \
for jail test_02.

test_02.
```

```
[root@test_23 /home/admin]# ezjail-admin stop test_03
Stopping jails:/etc/rc.d/jail: WARNING: /var/run/jail.test_03.conf is created and used \
for jail test_03.

test_03.
```

```
[root@test_23 /home/admin]# ezjail-admin list
```

STA	JID	IP	Hostname	Root Directory
ZS	N/A	127.0.2.3	test_03	/local/jails/test_03
		N/A em0 10.1.0.53		
ZS	N/A	127.0.2.2	test_02	/local/jails/test_02
		N/A em0 10.1.0.52		
ZS	N/A	127.0.2.1	test_01	/local/jails/test_01
		N/A em0 10.1.0.51		

```
[root@test_23 /home/admin]# ezjail-admin archive -A
```

```
[root@test_23 /home/admin]# ls -l /export/archive/jails/ezjail_archives/
test_01-202403191152.32.tar.gz
test_02-202403191152.13.tar.gz
test_03-202403191151.55.tar.gz
```

```
Update jails' configuration data test_23/jails/jail.d/*.yaml
```

```
Start the jails
```

```
[root@test_23 /home/admin]# ezjail-admin start test_01
[root@test_23 /home/admin]# ezjail-admin start test_02
[root@test_23 /home/admin]# ezjail-admin start test_03
```

```
[root@test_23 /home/admin]# ezjail-admin list
```

STA	JID	IP	Hostname	Root Directory
ZR	6	127.0.2.3	test_03	/local/jails/test_03
	6	em0 10.1.0.53		
ZR	5	127.0.2.2	test_02	/local/jails/test_02
	5	em0 10.1.0.52		
ZR	4	127.0.2.1	test_01	/local/jails/test_01
	4	em0 10.1.0.51		

## Stop and delete jail

Modify the configuration of *test\_02*. Set `start: false` and `present: false`

```
(env) > cat test_23/jails/jail.d/test_02.yml
```

---

```
objects:
- jailname: test_02
  present: false
  start: false
  jailtype: zfs
  ...
```

(abridged)

Delete the jail *test\_02*

```
(env) > ansible-playbook test_23/freebsd-jail.yml -t bsd_jail_ezjail_jails
```

```
PLAY [test_23] *****
```

```
TASK [vbotka.freebsd_jail : ezjail-jails: Delete jails] *****
```

```
included: /home/admin/.ansible/roles/vbotka.freebsd_jail/tasks/fn/delete.yml \
  for test_23 => (item=test_02)
```

```
included: /home/admin/.ansible/roles/vbotka.freebsd_jail/tasks/fn/delete.yml \
  for test_23 => (item=test_04)
```

```
TASK [vbotka.freebsd_jail : ezjail-jails:delete: Shutdown jail test_02 id 5] ***
changed: [test_23]
```

```
TASK [vbotka.freebsd_jail : ezjail-jails:delete: Stop jail test_02] *****
changed: [test_23]
```

```
TASK [vbotka.freebsd_jail : ezjail-jails:delete: Delete jail test_02] *****
changed: [test_23]
```

```
TASK [vbotka.freebsd_jail : ezjail-jails:delete: Delete stamp \
  /var/db/jail-stamps/test_02-firstboot] *****
changed: [test_23]
```

```
TASK [vbotka.freebsd_jail : ezjail-jails: Create and configure jails] *****
```

```
included: /home/admin/.ansible/roles/vbotka.freebsd_jail/tasks/fn/create.yml \
  for test_23 => (item=test_01)
```

```
included: /home/admin/.ansible/roles/vbotka.freebsd_jail/tasks/fn/create.yml \
  for test_23 => (item=test_03)
```

```
PLAY RECAP *****
```

```
test_23: ok=29 changed=4 unreachable=0 failed=0 skipped=27 rescued=0 ignored=0
```

Take a look at the list of jails

STA	JID	IP	Hostname	Root Directory
ZR	6	127.0.2.3	test_03	/local/jails/test_03
	6	em0 10.1.0.53		
ZR	4	127.0.2.1	test_01	/local/jails/test_01
	4	em0 10.1.0.51		

### Restore and start jail

Modify the configuration of *test\_02*. Set `start: true` and `present: true`

```
(env) > cat test_23/jails/jail.d/test_02.yml
```

```
---
objects:
  - jailname: test_02
    present: true
    start: true
    jailtype: zfs
    archive: test_02-202403191152.13.tar.gz
    ...
```

(abridged)

Create the jail *test\_02*. If the jail does not exist it will be restored from the archive if the parameter *bsd\_ezjail\_admin\_restore* is set *true* (default is *false*).

Speedup the play by selecting the tag *bsd\_jail\_ezjail\_jails*

```
(env) > ansible-playbook test_23/freebsd-jail.yml -e bsd_ezjail_admin_restore=true \
-t bsd_jail_ezjail_jails
```

```
PLAY [test_23] *****
```

```
TASK [vbotka.freebsd_jail : ezjail-jails: Delete jails] *****
```

```
included: /home/admin/.ansible/roles/vbotka.freebsd_jail/tasks/fn/delete.yml \
for test_23 => (item=test_04)
```

```
TASK [vbotka.freebsd_jail : ezjail-jails: Create and configure jails] *****
```

```
included: /home/admin/.ansible/roles/vbotka.freebsd_jail/tasks/fn/create.yml \
for test_23 => (item=test_01)
```

```
included: /home/admin/.ansible/roles/vbotka.freebsd_jail/tasks/fn/create.yml \
for test_23 => (item=test_02)
```

```
included: /home/admin/.ansible/roles/vbotka.freebsd_jail/tasks/fn/create.yml \
for test_23 => (item=test_03)
```



```
TASK [vbotka.freebsd_jail : ezjail-jails:create: Create or Restore jail test_02] **
changed: [test_23]
```

```
TASK [vbotka.freebsd_jail : ezjail-jails:create: \
Configure parameters /usr/local/etc/ezjail/test_02] ***
changed: [test_23]
```

```
PLAY RECAP *****
test_23: ok=33 changed=2 unreachable=0 failed=0 skipped=35 rescued=0 ignored=0
```

Take a look at the list of jails

```
[root@test_23 /home/admin]# ezjail-admin list
```

STA	JID	IP	Hostname	Root Directory
ZR	6	127.0.2.3	test_03	/local/jails/test_03
	6	em0 10.1.0.53		
ZS	N/A	127.0.2.2	test_02	/local/jails/test_02
	N/A	em0 10.1.0.52		
ZR	4	127.0.2.1	test_01	/local/jails/test_01
	4	em0 10.1.0.51		

The jail *test\_02* is not running. The status is *ZS* and *JID* is missing. Start the jail

```
(env) > ansible-playbook test_23/freebsd-jail.yml -t bsd_jail_start
```

```
PLAY [test_23] *****
```

```
TASK [vbotka.freebsd_jail : start: Start jails] *****
changed: [test_23] => (item=test_02)
```

```
PLAY RECAP *****
test_23: ok=4 changed=1 unreachable=0 failed=0 skipped=2 rescued=0 ignored=0
```

Take a look at the list of jails. The jail *test\_02* is running again

```
[root@test_23 /home/admin]# ezjail-admin list
```

STA	JID	IP	Hostname	Root Directory
ZR	6	127.0.2.3	test_03	/local/jails/test_03
	6	em0 10.1.0.53		
ZR	7	127.0.2.2	test_02	/local/jails/test_02
	7	em0 10.1.0.52		
ZR	4	127.0.2.1	test_01	/local/jails/test_01
	4	em0 10.1.0.51		

## Troubleshooting

Optionally, get the script `my-jail-admin.sh` from the Ansible role `freebsd_jail` `contrib/bin`

```
shell> scp roles/vbotka.freebsd_jail/contrib/bin/my-jail-admin.sh admin@test_23:~
```

For example, display the status of the jail `test_02`. This can help with debugging of potential problems. The status of `test_02` displays the warning that `/var/db/jail-stamps/test_02-firstboot` does not exist

```
[root@test_23 /home/admin]# /home/admin/my-jail-admin.sh status test_02
[Logging: /tmp/my-jail-admin.test_02]
2024-03-19 14:20:59: test_02: status: [OK] pid: /var/run/jail_test_02.id
2024-03-19 14:20:59: test_02: status: [OK] conf: /var/run/jail.test_02.conf
2024-03-19 14:20:59: test_02: status: [OK] jail: /local/jails/test_02
2024-03-19 14:20:59: test_02: status: [WRN] lock: /var/db/jail-stamps/test_02-firstboot does not exist
2024-03-19 14:20:59: test_02: status: [OK] fstab: /etc/fstab.test_02
2024-03-19 14:20:59: test_02: status: [OK] fstab: /etc/jail/fstab.test_02
2024-03-19 14:20:59: test_02: status: [OK] conf: /etc/jail.conf
2024-03-19 14:20:59: test_02: status: [OK] conf: /usr/local/etc/ezjail/test_02
2024-03-19 14:20:59: test_02: status: [OK] jail_rcd:

  JID      IP Address      Hostname      Path
  test_01   127.0.2.1       test_01       /local/jails/test_01
  test_03   127.0.2.3       test_03       /local/jails/test_03
  test_02   127.0.2.2       test_02       /local/jails/test_02
```

## Freebsd jails as Ansible remote hosts

Update the inventory

```
shell> cat hosts
```

```
[test]
test_23
```

```
[test:vars]
ansible_python_interpreter=/usr/local/bin/python3.9
```

```
[jails]
test_01
test_02
test_03
```

```
[jails:vars]
ansible_python_interpreter=/usr/local/bin/python3.9
```

Create a playbook to test the connection

```
shell> cat jails/test.yml
```

```

---
- name: Test connection to the jails
  hosts: jails
  remote_user: admin

  tasks:

    - command: hostname
      register: out

    - debug:
        var: out.stdout

```

Test the connection

```
(env) > ansible-playbook jails/test.yml
```

```

PLAY [Test connection to the jails] *****

TASK [Gathering Facts] *****
ok: [test_01]
ok: [test_02]
ok: [test_03]

TASK [command] *****
changed: [test_03]
changed: [test_02]
changed: [test_01]

TASK [debug] *****
ok: [test_01] =>
  out.stdout: test_01
ok: [test_02] =>
  out.stdout: test_02
ok: [test_03] =>
  out.stdout: test_03

PLAY RECAP *****
test_01: ok=3 changed=1 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
test_02: ok=3 changed=1 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
test_03: ok=3 changed=1 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

```

## Ansible module iocage

See the README of the Ansible module iocage.

## **Legal notice**

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

## **License**

CC BY-SA 4.0 ATTRIBUTION-SHAREALIKE 4.0 INTERNATIONAL

---