

# FreeBSD Jails Orchestration with Ansible

## Tutorial with examples

Vladimir Botka

# Contents

- 1 Introduction
- 2 FreeBSD collection
- 3 Examples
- 4 Install, configure, and activate iocage
- 5 Plugins iocage
- 6 Ansible client
- 7 Infrastructure

# Section 1

## Introduction

# Why Ansible?

- Preferred for its simplicity due to an agentless model using YAML.
- Used by global leaders (AWS, Cisco, Google, ...).
- IBM and Red Hat Ansible are closely integrated, particularly through the Red Hat Ansible Automation Platform and the generative AI service, Red Hat Ansible Lightspeed with IBM watsonx Code Assistant.

# Does Ansible work with FreeBSD?

Yes. Quoting Ansible documentation [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. Please feel free to report any issues or incompatibilities you discover with BSD; pull requests with an included fix are also welcome!"

See Ansible documentation [Managing BSD hosts with Ansible](#).

# Ansible collections

- There are more than 100 collections included in the Ansible distribution.

```
shell> ansible-galaxy collection list | wc -l  
107
```

- For example:
  - ▶ [Amazon.Aws](#)
  - ▶ [Cisco.Ios](#)
  - ▶ [Google.Cloud](#)
- See the [Collection Index](#)

# Ansible collections ansible.\*

Tested with FreeBSD:

- **ansible.builtin** - Modules and plugins contained in ansible-core.  
author: Ansible, Inc
- **ansible.posix** - For POSIX and POSIX-ish platforms.  
author: Ansible (github.com/ansible)
- **ansible.utils** - Data management, manipulation, and validation.  
author: Ansible Community

See the release notes what FreeBSD version(s) were tested. For example, quoting **v2.20.0**:

```
ansible-test - Replace FreeBSD 14.2 with 14.3.
```

# Ansible collections community.\*

- There are two FreeBSD specific modules in `community.general`:
  - ▶ `community.general.pkgng` - Package manager for FreeBSD.
  - ▶ `community.general.portinstall` - Installing from FreeBSD's ports system.
- See other collections `community.*` For example:
  - ▶ `community.crypto` - Modules and plugins for cryptographic operations.
  - ▶ `community.postgresql` - PostgreSQL community modules.
  - ▶ `community.mysql` - MySQL and MariaDB collection.
- These collections are maintained by the Ansible community.



# Ansible collection dedicated to FreeBSD is needed to

- support FreeBSD specific subsystems:
  - ▶ iocage
  - ▶ poudriere
  - ▶ bhyve
- support FreeBSD plugins where integration is problematic:
  - ▶ service
  - ▶ sysctl

## Section 2

### FreeBSD collection

# Proposed FreeBSD collection

- Ansible Galaxy:  
<https://galaxy.ansible.com/ui/repo/published/vbotka/freebsd/>
- GitHub:  
<https://github.com/vbotka/ansible-collection-freebsd/>
- Read The Docs:  
<https://ansible-collection-freebsd.readthedocs.io/en/latest/>

# Collection Content

- **Plugins:**
  - ▶ module iocage - iocage jail handling.
  - ▶ module service - Control or list system services.
  - ▶ module ucl - CRUD-like interface for managing UCL files.
  - ▶ inventory iocage - iocage inventory source.
  - ▶ filter iocage - Parse iocage lists.
  - ▶ lookup galaxy\_info - Get the meta data from galaxy.yml
- **Roles**
- **Playbooks**

Note: The proposed FreeBSD collection is a work in progress.

# We focus on the iocage plugins

- **inventory iocage** - iocage inventory source.
- **module iocage** - iocage jail handling.
- **filter iocage** - Parse iocage lists.

# inventory iocage

- Get inventory hosts from the `iocage` jail manager running on `host`.
- Properly documented in [ansible-collection-freebsd.readthedocs.io](https://docs.ansible.com/ansible-2.9/collections/community/general/iocage.html)
- Included also in the collection `community.general` as [community.general.iocage](https://docs.ansible.com/ansible-2.9/collections/community/general/iocage.html)
- Also documented in [docs.ansible.com](https://docs.ansible.com/)
- License GPLv3
- Extends fragments:

`extends_documentation_fragment:`

- `ansible.builtin.constructed`
- `ansible.builtin.inventory_cache`

# inventory iocage - Basics

- Get the inventory from the command line

```
shell> ssh admin@10.1.0.73 iocage list -l
```

- Configure inventory plugin hosts/02\_iocage.yml

```
plugin: vbotka.freebsd.iocage  
host: 10.1.0.73  
user: admin
```

- Display the inventory

```
shell> ansible-inventory -i hosts/02_iocage.yml --list --yaml
```

## inventory iocage - DHCP

- As admin at the controller, list the jails. The IP4 tab says “... address requires root”. Use sudo if enabled

```
shell> ssh admin@10.1.0.73 sudo iocage list -l
```

- Update the inventory configuration hosts/02\_iocage.yml. Use the parameter sudo

```
plugin: vbotka.freebsd.iocage
host: 10.1.0.73
user: admin
sudo: true
```

- Display the inventory

```
shell> ansible-inventory -i hosts/02_iocage.yml --list --yaml
```



## inventory iocage - Hooks

- When `sudo` is not enabled, update the inventory configuration `hosts/02_iocage.yml`. Use the parameter `hooks_results`

```
plugin: vbotka.freebsd.iocage
host: 10.1.0.73
user: admin
hooks_results:
  - /var/db/dhclient-hook.address.epair0b
```

- Example of a hook `/etc/dhclient-exit-hooks`

```
case "$reason" in
  "BOUND"|"REBIND"|"REBOOT"|"RENEW")
    echo $new_ip_address > /var/db/dhclient-hook.address.$interface
    ;;
esac
```

- Display the inventory

```
shell> ansible-inventory -i hosts/02_iocage.yml --list --yaml
```

## inventory iocage - Properties

Optionally, get the `iocage` properties. Update the inventory configuration `hosts/02_iocage.yml`. Use the parameter `get_properties`. Compose the variable `ansible_host`

```
plugin: vbotka.freebsd.iocage
host: 10.1.0.73
user: admin
get_properties: true
hooks_results:
  - /var/db/dhclient-hook.address.epair0b
compose:
  ansible_host: (iocage_hooks.0 == '-') |
                ternary(iocage_ip4, iocage_hooks.0)
```

## inventory iocage - Tags

Use property notes to store tags. Update the inventory configuration hosts/02\_iocage.yml. Compose the dictionary iocage\_tags and create groups. The properties are required. Enable the parameter get\_properties

```
plugin: vbotka.freebsd.iocage
host: 10.1.0.73
user: admin
get_properties: true
hooks_results:
  - /var/db/dhclient-hook.address.epair0b
compose:
  ansible_host: (iocage_hooks.0 == '-') |
                ternary(iocage_ip4, iocage_hooks.0)
  iocage_tags: dict(iocage_properties.notes |
                  split | map('split', '='))
keyed_groups:
  - prefix: vmm
    key: iocage_tags.vmm
  - prefix: project
    key: iocage_tags.project
```

## inventory iocage - Aliases

The tag `alias` is used to create inventory aliases. Update the inventory configuration `hosts/02_iocage.yml`. Set the parameter `inventory_hostname_tag` to `alias`. This tag keeps the value of the inventory alias.

```
plugin: vbotka.freebsd.iocage
host: 10.1.0.73
user: admin
get_properties: true
inventory_hostname_tag: alias
hooks_results:
  - /var/db/dhclient-hook.address.epair0b
compose:
  ansible_host: (iocage_hooks.0 == '-') |
                ternary(iocage_ip4, iocage_hooks.0)
  iocage_tags: dict(iocage_properties.notes | split | map('split', '='))
keyed_groups:
  - prefix: vmm
    key: iocage_tags.vmm
  - prefix: project
    key: iocage_tags.project
```

# module iocage

- FreeBSD iocage jail handling.
- See the in-line documentation on [galaxy.ansible.com](https://galaxy.ansible.com)
- Implements most of the `iocage` options.
- See the in-line examples.

## filter iocage

- This filter parses `iocage` list output.
- See the in-line documentation on [galaxy.ansible.com](https://galaxy.ansible.com)
- See the in-line examples.

# Section 3

## Examples

# Example groups

- Install, configure, and activate `iocage`
- Plugins `iocage`
- Ansible client
- Infrastructure



# How to use the examples

Each example provides links to the:

- example source code
- detailed description of the example
- example results

To test the examples:

- clone the source code repository
- open the guide with the detailed descriptions
- run the examples and compare the results

# Notes

- All examples comprise additional files not shown in the file' tree. See them for more details.
- Most examples comprise `batch.sh` that runs the commands and creates the output.
- Most plays in `batch.sh` are idempotent. The output of such a play may show status `ok` instead of expected `changed` if the play has already been run.
- The playbooks in the examples use dashes - in their filenames. For example, `pb-iocage.yml`.
- The playbooks in the collection, because of the Ansible collection naming conventions, use underscores `_` in their filenames. For example, `pb_iocage_template.yml`.

## Section 4

Install, configure, and activate iocage

## example 001: Install iocage

Use the role `vbotka.freebsd.iocage` to install the package `iocage`.

*requirements:*

- root privilege in the managed nodes

*links:*

- [source code](#)
- [results](#)

## example 002: Activate iocage

Use the role `vbotka.freebsd.iocage` to activate iocage.

*requirements:*

- root privilege in the managed nodes
- binary iocage

*links:*

- [source code](#)
- [results](#)

## example 003: Audit iocage host

Use the role `vbotka.freebsd.iocage` to audit the iocage configuration.

*requirements:*

- root privilege in the managed nodes
- binary iocage

*links:*

- [source code](#)
- [results](#)

## Section 5

### Plugins iocage

## example 010: Clone basejails and create inventory

Fetch releases, create basejails, clone jails from the basejails, and start the jails. Use the inventory plugin `vbotka.freebsd.iocage` to create the inventory. Display the created inventory.

*requirements:*

- module `vbotka.freebsd.iocage`
- inventory plugin `vbotka.freebsd.iocage`
- root privilege in the managed nodes
- activated binary `iocage`

*links:*

- [source code](#)
- [results](#)



## example 011: Display variables `iocage_*`

*extends:* example 010

Display all variables `iocage_*` created by the inventory plugin `vbotka.freebsd.iocage`.

*requirements:*

- inventory plugin `vbotka.freebsd.iocage`
- jails created in example 010

*links:*

- [source code](#)
- [results](#)

## example 012: Display iocage\_properties

*extends:* example 010

Enable and display `iocage_properties`.

*requirements:*

- inventory plugin `vbotka.freebsd.iocage`
- jails created in example 010

*links:*

- [source code](#)
- [results](#)

## example 013: Tags and custom groups

*extends:* example 010

Use the property **notes** to create tags:

- Add the property **notes**: "vmm=localhost"

In the inventory plugin:

- compose the variable **iocage\_tags**
- create groups **vmm\_\*** from the attribute **iocage\_tags.vmm**

*requirements:*

- module **vbotka.freebsd.iocage**
- inventory plugin **vbotka.freebsd.iocage**
- root privilege in the managed nodes
- activated binary **iocage**
- fetched releases

*links:*

- [source code](#)
- [results](#)

## example 014: Inventory cache

Enable and test inventory cache.

*requirements:*

- inventory plugin `vbotka.freebsd.iocage`
- jails created in example 010

*links:*

- [source code](#)
- [results](#)

## example 015: Multiple inventory cache

Enabled cache in multiple inventory files.

*requirements:*

- inventory plugin `vbotka.freebsd.iocage`
- jails created in example 010

*links:*

- [source code](#)
- [results](#)

## example 016: Multiple inventory constructed

Create inventory groups using the inventory plugin `ansible.builtin.constructed` after the two inventory plugin `vbotka.freebsd.iocage` configuration files.

*requirements:*

- inventory plugin `vbotka.freebsd.iocage`
- jails created in example 010

*links:*

- [source code](#)
- [results](#)

## example 017: `community.general.iocage`

Use the inventory plugin `community.general.iocage` instead of the inventory plugin `vbotka.freebsd.iocage`.

*links:*

- [source code](#)
- [results](#)

## example 018: Clone basejails. Use DHCP.

Use DHCP to configure the interfaces.

*requirements:*

- inventory plugin `vbotka.freebsd.iocage`
- jails created in example 010

*links:*

- [source code](#)
- [results](#)



## example 019: Inventory option `use_vars_plugins`

The option `use_vars_plugins`, responsible for reading `host_vars` and `group_vars` directories, is not available in the inventory plugin `vbotka.freebsd.iocage` because the `constructed` fragment doesn't provide it.

- Use the inventory plugin `ansible.builtin.constructed` to read `group_vars`.
- Use the variable `region` to create the groups `region_EU` and `region_US`.

*links:*

- [source code](#)
- [results](#)

## example 020: Get inventory aliases from notes

Get the inventory aliases from the `iocage` property notes. In the inventory plugin `vbotka.freebsd.iocage`, use the option `inventory_hostname_tag` to tell the plugin which tag to use.

*requirements:*

- inventory plugin `vbotka.freebsd.iocage`
- root privilege in the managed nodes
- templates created in example 202

*links:*

- [source code](#)
- [results](#)

## example 030: Create custom facts

*extends:* example 020

Create custom facts to provide a dictionary of iocage datasets lists. Use the filter `vbotka.freebsd.iocage` to parse them.

*requirements:*

- role `vbotka.freebsd.iocage`
- filter `vbotka.freebsd.iocage`
- root privilege in the managed nodes
- jails created in previous examples

*links:*

- [source code](#)
- [results](#)

## Section 6

### Ansible client

## example 200: Create iocage templates. Clone jails.

- Create iocage templates for Ansible clients.
- Clone jails.

### *requirements:*

- `playbook vbotka.freebsd.pb_iocage_template.yml`
- `playbook vbotka.freebsd.pb_iocage_ansible_clients.yml`
- `module vbotka.freebsd.iocage`
- `inventory plugin vbotka.freebsd.iocage`
- root privilege in the managed nodes
- activated `iocage`
- fetched releases

### *links:*

- [source code](#)
- [results](#)

## example 201: Display iocage datasets

- Get and display iocage datasets.

*links:*

- [source code](#)
- [results](#)

## example 202: Create iocage templates. Clone DHCP jails

*extends:* example 200

- Create iocage templates for Ansible clients.
- Get the IP addresses by DHCP.
- Create the `dhclient-exit-hooks`.

*requirements:*

- playbook `vbotka.freebsd.pb_iocage_template.yml`
- playbook `vbotka.freebsd.pb_iocage_ansible_clients.yml`
- module `vbotka.freebsd.iocage`
- inventory plugin `vbotka.freebsd.iocage`
- root privilege in the managed nodes
- activated `iocage`
- fetched releases

*links:*

- [source code](#)
- [results](#)

## example 203: Create DHCP jails with auto UUID and iocage\_tags

- Automatically generate the jails UUID names. At each iocage host, create three jails from the template `ansible_client`. The module `vbotka.freebsd.iocage` doesn't work with multiple names. Use `ansible.builtin.command` instead.
- In the inventory plugin, compose the variable `iocage_tags`  
`iocage_tags: dict(iocage_properties.notes | split | map('split', '='))`
- Create groups from `iocage_tags`

### *requirements:*

- playbook `vbotka.freebsd.pb_iocage_ansible_clients.yml`
- module `vbotka.freebsd.iocage`
- inventory plugin `vbotka.freebsd.iocage`
- root privilege in the managed nodes
- templates created in example 202

### *links:*

- [source code](#)
- [results](#)



## example 204: Create DHCP jails with auto UUID and iocage\_tags v2

*extends:* example 203

- Instead of the module `vbotka.freebsd.iocage` create the variable `iocage_jails` using the filter `vbotka.freebsd.iocage`

*requirements:*

- filter `vbotka.freebsd.iocage`
- inventory plugin `vbotka.freebsd.iocage`
- root privilege in the managed nodes
- templates created in example 202

*links:*

- [source code](#)
- [results](#)

## example 206: Create DHCP and fixed IP jails

*extends:* example 203

- In the inventory plugin `vbotka.freebsd.iocage` configuration file, use the option `hooks_results` to get the DHCP IP address.

*requirements:*

- playbook `vbotka.freebsd.pb_iocage_ansible_clients.yml`
- module `vbotka.freebsd.iocage`
- inventory plugin `vbotka.freebsd.iocage`
- root privilege in the managed nodes
- templates created in example 202

*links:*

- [source code](#)
- [results](#)

## example 207: Create DHCP jails with auto UUID, iocage\_tags, alias and class

- At multiple iocage hosts, create and run VNET jails with a DHCP interface from the template `ansible_client`.
- Use the dictionary `iocage_tags` and option `inventory_hostname_tag` to create `inventory aliases`.
- Group the jails by iocage hosts, states, and classes.
- Declare the project in a single dictionary. The dictionary keys are jails' aliases.

### *requirements:*

- inventory plugin `vbotka.freebsd.iocage`
- root privilege in the managed nodes
- templates created in example 202

### *links:*

- [source code](#)
- [results](#)

## example 208: Create iocage template for ansible-pull

- Create iocage template `ansible_client_pull` that will use `ansible-pull`.

*requirements:*

- playbook `vbotka.freebsd.pb_iocage_template.yml`
- module `vbotka.freebsd.iocage`
- root privilege in the managed nodes

*links:*

- [source code](#)
- [results](#)

## example 209: Create iocage pkglist file

- Use the role `vbotka.freebsd.iocage` to create `iocage__` list of packages for Automatic Package Installation.
- Create Ansible template for Apache HTTP server.

*requirements:*

- role `vbotka.freebsd.iocage.yml`
- playbook `vbotka.freebsd.pb_iocage_template.yml`

*links:*

- [source code](#)
- [results](#)

## example 210: Test empty iocage notes

- Test empty iocage notes. Create `iocage_tags`. The result should be an empty dictionary.

*requirements:*

- inventory plugin `vbotka.freebsd.iocage`

*links:*

- [source code](#)
- [results](#)

# Section 7

## Infrastructure

## example 500: syslog-ng server and syslog-ng clients

Configure and run a log server. Configure log clients and test them. Use **syslog-ng**. Use the jails created in the example 207 Create DHCP jails with `auto UUID`, `iocage_tags`, `alias` and `class`. The project keys are jail's aliases.

*requirements:*

- inventory plugin `vbotka.freebsd.iocage`
- module `vbotka.freebsd.service`
- role `vbotka.freebsd.postinstall`
- jails created in example 207

*links:*

- [source code](#)
- [results](#)



## example 501: iocage host

Configure iocage host.

*requirements:*

- `role vbotka.freebsd.iocage`
- `role vbotka.freebsd.network`
- `role vbotka.freebsd.pf`
- `role vbotka.freebsd.postinstall`
- `role vbotka.freebsd.zfs`

*links:*

- [source code](#)
- [results](#)

## example 502: Branch server

Install and configure syslog-ng and git servers in the branch-server.

*requirements:*

- `role vbotka.freebsd.config_light`
- `role vbotka.freebsd.postinstall`

*links:*

- [source code](#)
- [results](#)

## example 510: Project ansible-pull

Use the template `ansible_client_pull` to create project jails.

*requirements:*

- inventory plugin `vbotka.freebsd.iocage`
- root privilege in the managed nodes
- templates `ansible_client_pull` created in example 208

*links:*

- [source code](#)
- [results](#)