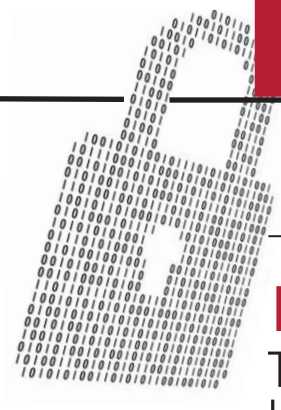


iocage

The One True Jail Manager

by Michael W Lucas

The title of this article isn't exactly a lie, but it's not the complete truth.

Other jail managers exist. Some work quite serviceably in specific-use cases. But for medium-to-large jail servers, where you expect to have dozens or hundreds of jails on a single host, iocage increases manageability. Iocage can go up against docker. Iocage leverages ZFS. Iocage automates mass upgrades of jails, including packages. Iocage configures virtual networking. Iocage solves problems you didn't even know you had. Iocage is always spelled with a lower case "I," but that looks silly at the beginning of a sentence, so I refuse.

If you're deploying a jail host today: use iocage. (And use FreeBSD 12 or later, but that's a separate topic.) Yes, iocage requires ZFS and Python. A jail host these days should use ZFS, and Python is pretty ubiquitous. If you have enough disks in your host, put the host operating system on one redundant ZFS pool and your jails on another. Lock down the host's services; then install iocage from package or Github.

First, tell iocage which pool to use for jails.

```
# iocage activate jails
```

If you don't specify a pool, iocage puts all the jails on the root pool.

The iocage Command Line

Perform all actions with the iocage(8) command. The syntax strongly resembles that of ZFS, as if the iocage developers knew a good idea when they saw it. We'll start by viewing iocage's default parameters with **iocage get**. We add the **-a** flag to get all parameters and add the jail name *default* to see the default parameters.

```
# iocage get -a default

CONFIG_VERSION:20
allow_chflags:0
allow_mlock:0
allow_mount:0
…
```

Parameters in iocage are delimited by underscores instead of periods, because of Python. The parameter *allow_chflags* is identical to the jail.conf *allow.chflags* parameter. With *allow_chflags* set to 0, iocage doesn't set allow.chflags by default.

As iocage uses parameters to configure absolutely everything about a jail, some parameters have no jail.conf equivalent. The *resolver* parameter tells iocage where to get a resolv.conf for the jail.

```
…
resolver:/etc/resolv.conf
…
```

The jail copies /etc/resolv.conf from the host. Change the setting with the iocage set command. You could name a specific jail or change the default to have it take effect for all jails created from now on.

```
# iocage set resolver=/etc/resolv.conf.jail default
```

If the defaults seem sensible, create your first jail.

You'll need a FreeBSD release before you can create a jail. Use the iocage fetch command to see what's available.

```
# iocage fetch
[0] 11.2-RELEASE
[1] 11.3-RELEASE
[2] 12.0-RELEASE
```

Type the number of the desired RELEASE.

Press [Enter] to fetch the default selection: (12.0-RELEASE). It defaults to grabbing the newest release, but you can choose an earlier one if you wish. The selected release is downloaded, extracted on local disk, and updated with all relevant security patches so that your jails can clone it for their filesystem.

View all the already downloaded releases with `iocage list -r`.

Creating Jails and Setting Parameters

You can create a jail without a hostname or IP address, but iocage will assign it a random UUID instead of a useful name and the jail won't get network access. Specify a jail name with -n, and also set the ip4_addr or ip6_addr parameter to give it networking. Choose a release with `-r`.

```
# iocage create -n wwww1 ip4_addr="203.0.113.234" -r 11.2-RELEASE
wwww1 successfully created!
```

Iocage ZFS-clones the chosen release for the jail. This makes jail creation really fast but makes new jails deceptively small. If you expect this jail to be around a long time, you might choose to make a thick jail by adding the `-T` flag.

Change a jail name with `iocage rename`.

```
# iocage rename wwww1 www1
```

Remove jails with `iocage destroy`.

```
# iocage destroy www1
```

You can now run the jails.

iocage Startup and Shutdown

Like any sensible jail manager, iocage assumes that jails shouldn't automatically run at system boot. Use the boot parameter to control jail boot at startup. Use `iocage get -r` to recursively grab a parameter's value from all jails.

```
# iocage get -r boot
+------+-------------+
| NAME | PROP - boot |
+======+=============+
| www1 | 0           |
+------+-------------+
| www2 | 0           |
+------+-------------+
```

Neither of these jails will automatically run at system startup. Change a jail's boot parameter to `on,yes,` or `1` to have it start at boot.

```
# iocage set boot=on www1
boot: 0 -> 1
```

Start, stop, and reboot jails with the `start,stop`, and `restart` commands. Give either a jail name or use `ALL` to affect all jails.

```
# iocage start ALL
```

Viewing Jails

While you can use standard commands like jls(8) to view jails, you can grab iocage-specific information from your running jails with `iocage list`.

```
# iocage list
+-----+------+-------+--------------+------------------+
| JID | NAME | STATE |    RELEASE   |        IP4       |
+=====+======+=======+==============+==================+
| -   | www1 | down  | 12.0-RELEASE | 203.0.113.234/24 |
+-----+------+-------+--------------+------------------+
| -   | www2 | down  | 12.0-RELEASE | 203.0.113.235/24 |
+-----+------+-------+--------------+------------------+
```

…

For a list with more information, including IPv6 and template information, add the `-l flag`.

iocage Packages

Use the iocage pkg subcommand to manage packages. All package functions, including upgrades, work with iocage. Give the jail name, the pkg(8) command, and a package name.

```
# iocage pkg www1 install sudo
```

You can also use `pkg -j` to manage jail packages from the host. I recommend you pick one method and stick with it, however.

iocage Templates

Iocage lets you create a pristine model jail, with all of your packages and configuration files, and use that jail as a template for other jails. This template jail is a ZFS clone of a FreeBSD release, and the template gets cloned in turn for other jails. Your jails need LDAP and Kerberos? Set them up once and never again.

Create your jail exactly like any other jail. Once your model jail is perfect, set the jail's template property.

```
# iocage set template=yes wwwtemplate
```

Template jails cannot be started. View all your templates with `iocage list -t`.

To create a jail based on a template, use the `-t` flag to `iocage create`.

```
# iocage create -t wwwtemplate -n www2
```

You can assign the boot, ip4_addr, and ip6_addr properties at creation, or afterwards with a separate `iocage set` command.

iocage Plugins

All of the above features are nice, but they're not why I recommend iocage.

I recommend iocage because of plugins.

A plugin is a preconfigured jail that serves a single task. You want a jail to run antivirus? Grab the ClamAV plugin. A BitTorrent client? The qbittorrent plugin. You'll find several media servers, personal clouds, video camera managers, and more. While anyone can create a plugin, the official plugins are sanity-checked by the iocage team before being accepted into the repository. You won't find official plugins with everything running as root, or running extra daemons that transmit your login credentials back to the plugin author, or whatever daftness shows up in the Docker swamp.

To view all available plugins, run `iocage list -PR.`

```
# iocage list -PR
```

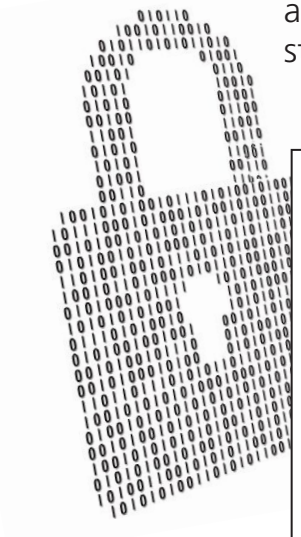
You'll get a list of all current plugins. Trawling through the list, I see there's a plugin for a UniFi controller. I have a UniFi wireless access point that piggybacks off a friend's controller, but it would be nice to have my own. According to `iocage list -PR`, the plugin is called `unificontroller`. Let's set this up. Use the iocage fetch command but add `-P` to indicate we're grabbing a plugin. Give the name of the plugin with `-n`.

```
# iocage fetch -P -n "unificontroller" ip6_addr=2001.db8::9/64
Plugin: unificontroller
  Official Plugin: True
  Using RELEASE: 11.2-RELEASE
  Using Branch: 12.0-RELEASE
  Post-install Artifact: https://github.com/lbalker/iocage-plugin-
unificontroller.git
  These pkgs will be installed:
    - unifi5
…
```

Iocage grabs the plugin's underlying release if you don't have it already available, install the packages and any configuration files, and configure it with the given IP address.

Using the plugin saves me the trouble of fiddling with the basics of getting the software running. I might have to configure it for my wacky local environment, but that's my own fault.

While this should be enough to get you started with iocage, it has all sorts of features we haven't touched. Base jails, ZFS delegation, and more are available in iocage. After you've used it for a few days, you'll understand why iocage is the One True Jail Manager. ●



MICHAEL W LUCAS is the author of ***Absolute FreeBSD***, ***FreeBSD Mastery: Jails***, and a bunch of other books. The new edition of ***Sudo Mastery*** should be out shortly after you read this, and his crime thriller ***Terrapin Sky Tango*** just escaped.

See the whole heap at https://mwl.io.