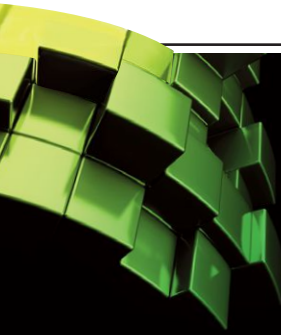


ADVANCED Jail Management WITH EZJAIL

by Andrew Fengler



Jails are very powerful tools for the modern system administrator. They allow lightweight **CONTAINERIZATION**, making it possible for you to easily isolate services and keep your physical hosts cleaner.

At ScaleEngine, we use ezjail to manage our jails, because it's simple, easy to hook up to our configuration automation system, and it's battle tested. It also does not get in our way when we need to make our jails do some more advanced tasks.

Some have described ezjail as "a pile of aging shell scripts." This claim is factual and accurate. However, a pile of shell scripts, as unsophisticated as it may be, is simple, easy to understand and debug, has no dependencies, and has very few weird corner cases. It's also much more mature than any other solution, which is always a good bet when you value stability.

Ezjail is two shell scripts, ezjail-admin which is used to interact with the jails and the ezjail-admin RC script, which does the heavy lifting of starting and stopping the jails. There is a single file, `/usr/local/etc/ezjail.conf` that controls some default settings, and a config file for each jail, `/usr/local/etc/ezjail/jailname`.

Basic Configuration of a Jail

First, we need to install ezjail. You can install it from ports or packages, as `sysutils/ezjail`.

The default ezjail config settings are pretty sane; about the only thing you will need to change to get a usable setup is to enable ZFS, which is off by default. Put the following into your `/usr/local/etc/ezjail.conf`:

```
ezjail_use_zfs="YES"  
ezjail_use_zfs_for_jails="YES"  
ezjail_jailzfs="dozer/jails"
```



This will cause ezjail to create a new dataset for each jail under dozer/jails. Now, we can tell ezjail to install a basejail for us to use:

```
# ezjail-admin install -m
```

Note: the -m flag installs the man pages in the jail, because nothing is more frustrating than not being able to look up the argument order for ln.

This will create datasets for basejail and newjail. The basejail dataset is nullfs mounted into each jail to provide the base system and allow for easy updates by simply replacing the contents of basejail. The newjail dataset is copied into each new jail that is created to provide a complete working system. Since we have these, we can create our first jail:

```
# ezjail-admin create myjail.example.com 10.0.0.1
```

This command will create a jail called myjail.example.com, with the IP address of 10.0.0.1. You will need to make sure the address 10.0.0.1 is already bound to an interface. If you want ezjail to automatically bind the address, you can specify the interface with the address, separated with a pipe character (|):

```
# ezjail-admin create myjail.example.com 'mlxen0|10.0.0.1'
```

I like to name jails by their hostname, but you can use any name you like. Note that ezjail will replace full stops (.) and most other special characters with underscores (_) in the jails config file and a few other places where it uses the name of the jail.

We will then want to start the jail:

```
# ezjail-admin start myjail.example.com
```


We can now get a shell in our jail by using the `ezjail-admin console` command:

```
# ezjail-admin console myjail.example.com
```

And we're off to the races. You can see your list of jails with the `ezjail-admin list` command, start, stop, and restart them with start, stop, and restart subcommands respectively. We can also control whether the jail is set to run, with `ezjail-admin config`:

```
# ezjail-admin config -r {run|norun}
```

When you set a jail to 'norun', the utility prevents the jail from getting started with the sophisticated mechanism of renaming the config file for the jail to



have the '.norun' extension. This will also prevent you from starting the jail manually unless you prefix "one" on the start of your subcommands, i.e.

```
# ezjail-admin onestart myjail.example.com
```

Less Basic Configuration of a Jail

One of the most searched questions by people new to jails is "why can't I ping from my jail?" Ping requires the use of raw sockets, which is disabled by default for security. We should typically leave this disabled, but there are times when you need it, whether for debugging, or for something like Nagios that needs to be able to ping. Jails have a parameter, `allow.raw_sockets`, that is set to 0 by default. We can have ezjail set parameters for our jails with the aptly named 'parameters' option in the config file for the jail.

The ezjail config file for our jail is just a shell script that will get included by the startup script when it starts the jail. So, all of our jail settings are just lines like the following, in `/usr/local/etc/ezjail/myjail.example.com`:

```
export jail_myjail_example_com_parameters="allow.raw_sockets=1"
```

Note the conversion of `.` to `_`

These parameters are set when the jail starts, so if it is already running, we need to restart it to take effect.

A common situation for jails is that on a server with public IP addresses, you may have some programs that need Internet access, but you are either constrained on IP addresses, or do not wish to subject this program to the tender mercies of the public Internet. This is not difficult, unless the router for your server does not do NAT, as is the case with many dedicated server and colocation providers.

Since the router is not doing NAT, any jail with a private IP address will not be able to connect out. We can work around this by running our own NAT somewhere, but we probably do not want to NAT our entire server.

We can change the FIB, or routing table, for the jail. Make sure you have set `net.fibs` in `/boot/loader.conf` to a number higher than 1 before trying this. In your jail's config file, set:

```
export jail_myjail_example_com_fib="2"
```

If we set this to 2, then the jail will use FIB 2 after we [re]start the jail. We can configure FIB 2 to have whatever special routing our jail needs, without affecting the host.

Another powerful feature of both ZFS and jails is the ability to delegate a dataset to a jail. By delegating a dataset into a jail, the root user in the jail becomes able to create and destroy child datasets, adjust properties on the dataset, perform replication, and much more. This means we can create a storage setup that is isolated from the host operating system so that a runaway

script with elevated permissions can't break the host.

If we set the 'jailed' parameter on a dataset to on, the host is no longer able to mount or manage the dataset or any of its children as a security measure. Once we've set that parameter, we can now tell ezjail to delegate it to the jail by setting the following option:

```
export jail_myjail_example_com_zfs_datasets="dozer/customerfiles"
```

Now after we restart the jail, our hypothetical customer can manage the dataset and make use of all of ZFS's power without being able to mess up our host.

Jails are a fantastic tool that should be in every system administrator's kit, whether for the security, the compartmentalization, or just for how easy it makes it to keep your systems clean.

ANDREW FENGLER is a system administrator at ScaleEngine Inc., a video CDN. He is responsible for managing a world-spanning fleet of servers, mostly FreeBSD.

Jails are FreeBSD's Most Legendary Feature:

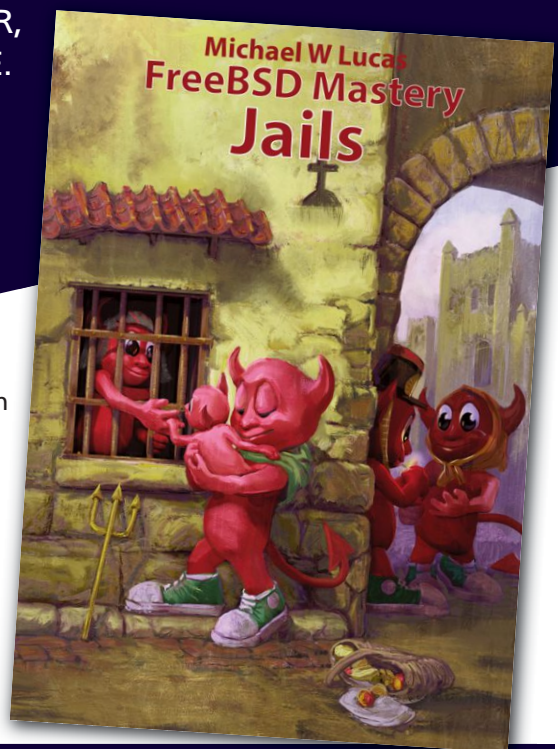
KNOWN TO BE POWERFUL, TRICKY TO MASTER,
AND CLOAKED IN DECADES OF DUBIOUS LORE.

FreeBSD Mastery: Jails cuts through the clutter to expose the inner mechanisms of jails and unleash their power in your service.

Confine Your Software!

- * Understand how jails achieve lightweight virtualization
- * Understand the base system's jail tools and the iocage toolkit
- * Optimally configure hardware
- * Manage jails from the host and from within the jail
- * Optimize disk space usage to support thousands of jails
- * Comfortably work within the limits of jails
- * Implement fine-grained control of jail features
- * Build virtual networks
- * Deploy hierarchical jails
- * Constrain jail resource usage
- ... **And much, much more!**

Available at Bookstores Everywhere



FreeBSD Mastery Jails BY MICHAEL W LUCAS