JAIL Migration

by Benedict Reuschling

Every once in a while, things needs to be moved around in your system. This is usually to make space for other things, or the new location is simply more fitting. This is true with simple files and directories, ZFS datasets, and sometimes whole installations. In my case, it was a jail that had started its life as a test system and I liked the installation so much that I did not want to set it all up again when the host it was running on was to be used for some other purpose. Luckily, I had set up the jail with iocage as the management framework. I knew that it supported migrations but had never used it before. Thus, it was a good opportunity to put my learning into an article.

Cold and Hot Migrations

There are two kinds of migrations: cold and hot. In a cold migration, the system or service is usually shut down for the time of the migration effort and then started again. A hot migration does not need that and can still provide all services and functions it normally provided. This is usually achieved using a shared medium, and even some clever networking tricks, to continue running established network connections. There are some variations where a hot migration happens so fast that the user does not realize it really was shut down briefly and then started again on the new location. Other methods, like a hot-standby system that takes over while one of the hosts is migrated, create the illusion of a constantly available service. These kinds of failover scenarios require planning and must be set up in this way (ideally from the start), which is usually not trivial.

After the cold migration is done and has been put in its new location, the jail can be started again. This process may sound trivial, but there are pitfalls. Most of them involve the fact that in the new environment, there will be a different network. New interface names and/or IP addresses need to be considered after the move.

In my case, it was a single jail running some services that were not critical enough to require the legendary 99.999% uptime. A little bit of downtime was acceptable and the data on the machine was not as big, so downtime would be minimal. With iocage, cold migrations are done by creating an archive of the ZFS datasets that make up the jail, plus a checksum to verify its integrity. As I mentioned earlier, the jail needs to be shut down to have a consistent archive without any open sockets or processes still running in the jail's main memory.

Exporting

To export a jail in iocage, identify the jail with its UUID by running:

```
# iocage list
+-----+--------+-------+-------------+-----------+
| JID |  NAME  | STATE |   RELEASE   |    IP4    |
+=====+========+=======+=============+===========+
| 1   | icinga | up    | 12.0-RELEASE | 10.0.0.15 |
+-----+--------+-------+-------------+-----------+
```

In this case, it is my icinga monitoring jail, and I can control it using the UUID provided in the NAME column. To export that jail, it needs to be shut down first. This is done with the iocage stop `<UUID>` command:

```
# iocage stop icinga
* Stopping icinga
  + Executing prestop OK
  + Stopping services OK
  + Tearing down VNET OK
  + Removing devfs_ruleset: 5 OK
  + Removing jail process OK
  + Executing poststop OK
```

Another call to iocage list should confirm that the jail is stopped now:

```
+-----+--------+-------+-------------+-----------+
| JID |  NAME  | STATE |   RELEASE   |    IP4    |
+=====+========+=======+=============+===========+
| -   | icinga | down  | 12.0-RELEASE | 10.0.0.15 |
+-----+--------+-------+-------------+-----------+
```

During the installation of iocage, a dataset called images is created among others on the pool dedicated for iocage. When executing the migration command, the archive making up the jail and the checksum are placed within the images directory.

```
# iocage export icinga
Exporting dataset: mypool/iocage/jails/icinga
Exporting dataset: mypool/iocage/jails/icinga/root
```

Preparing compressed file: /mypool/iocage/images/icinga_2019-10-10.zip. Depending on how much data is in the jail, the export process might take some time to finish. The same is true for the checksum of the file. As you can see from the output, the resulting zip file is named after the jail name plus the date of the export.

Copy the zip file to the new host. Once the copy operation is complete, make sure that the checksums still match with the checksum file generated in the export step. A simple way to verify the checksum is this:

```
# sha256 -r icinga_2019-10-10.zip
# cat icinga_2019-10-10.zip
```

That way, the two checksums are placed one over the other for an easy comparison. Next, install iocage on the new host if you have not done so already. After activating the pool (using iocage activate), create the directory structure using iocage fetch. Once it exists, the zip file must be placed in the images directory. From there, iocage can pick it up when running the import command:

```
# iocage import icinga
Importing dataset: icinga
Importing dataset: icinga/root

Imported: icinga
```

The jail will now show up in the iocage list output:

```
# iocage list
+-----+---------+-------+--------------+-----------+
| JID |  NAME   | STATE |   RELEASE    |    IP4    |
+=====+=========+=======+==============+===========+
| -   | icinga  | down  | 12.0-RELEASE | 10.0.0.15 |
+-----+---------+-------+--------------+-----------+
```

Depending on your new destination's jail setup, a new IP address must be set in order to allow networking for the jail just like before. By default, all the settings are preserved when migrating, including the old IP address, which may or may not be correct in the new location. Setting a new address might be as simple as using "iocage set ip4_addr" to a new IP address. Other options iocage supports are jails that share an IP and VNET jails. Check the iocage man page for more details. I can also recommend Michael W Lucas's book *FreeBSD Mastery: Jails*, which details a lot of different usage scenarios for jails with ready-to-use examples.

In my case, I was changing the IP address to a whole different network. Another run of iocage list will confirm that the settings were applied:

```
# iocage list
+-----+---------+-------+--------------+---------------+
| JID |  NAME   | STATE |   RELEASE    |      IP4      |
+=====+=========+=======+==============+===============+
| -   | icinga  | down  | 12.0-RELEASE | 192.168.1.128 |
+-----+---------+-------+--------------+---------------+
```

Starting the Jail

We can then start the jail using iocage start. Make sure that the jail in the old location is still in the down state, or both of them will vie for control of their IP address. Once the jail has started, make sure that the services running in it are listening to the new address. This usually includes changing the respective configuration files and then restarting the service to apply them. Once you are satisfied with the result and the jail is running in the new location, the old jail can be archived (backed up) or simply deleted. This frees up some space on the old host location. Don't forget the exported files in the iocage/images directory.

Jail migrations are not as scary as they may sound. With a bit of preparation and time to transfer the resulting files, it is a viable solution without having to set up jails completely from scratch again. •

Benedict Reuschling *joined the FreeBSD Project in 2009. After receiving his full documentation commit bit in 2010, he actively began mentoring other people to become FreeBSD committers. He joined the FreeBSD Foundation in 2015, where he is currently serving as vice president. Benedict has a Master of Science degree in Computer Science and is teaching a UNIX for software developers class at the Darmstadt University of Applied Sciences, Darmstadt, Germany. Together with Allan Jude, he is host of the weekly BSDNow.tv (http://BSDNow.tv) podcast.*