Bulk Ports Management with Poudriere

BY DREW GURKOWSKI

Step 1: Using the Ports Collection to Install Poudriere

All commands in this section should be run as root; using the `su(1)` command will work. To download a compressed snapshot of the Ports Collection into `/var/db/portsnap`:

```
# portsnap fetch
```

When running `portsnap` for the first time, extract the snapshot into `/usr/ports`:

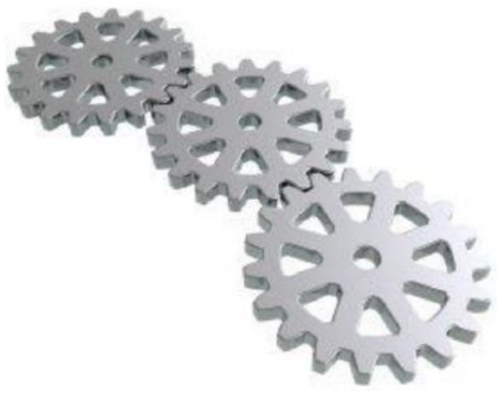
```
# portsnap extract
```

Next, build and install Poudriere from the Ports Collection using:

```
# cd /usr/ports/ports-mgmt/poudriere
# make install clean
```

Prompts will appear throughout the installation process, stick with the default settings and install the port.

Step 2: Configuring Poudriere



Poudriere is an extremely powerful tool designed for package production, but it can also be used to bulk manage ports. In order to do so, a few small tweaks will need to be made to the configuration before moving forward.

First, copy and move the configuration file to the correct location with:

```
# cd /usr/local/etc
# cp poudriere.conf.sample poudriere conf
```

Then use the `ee(1)` text editor to edit the copied configuration file:

```
# ee poudriere.conf
```

Arrow keys can be used to navigate down to the line:

```
FREEBSD_HOST=_PROTO_://_CHANGE_THIS_
```

Edit the above line, so it instead looks like this: (backspace to delete text)

```
FREEBSD_HOST=ftp://ftp.freebsd.org
```

Type `ESCAPE` and then `ENTER` twice to exit and save changes to the configuration file.

Step 3: Set Up a Poudriere Jail

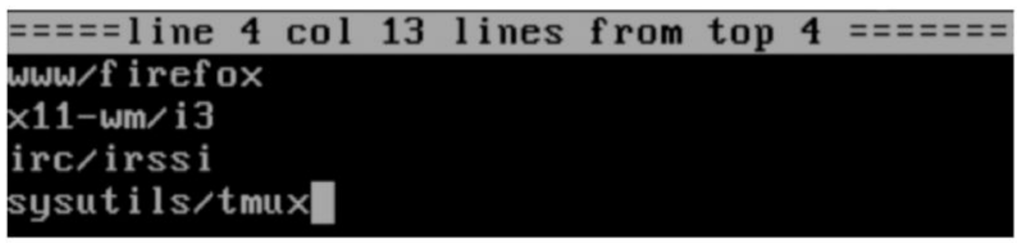
Before continuing, Poudriere will need to fetch and extract its own version of the FreeBSD Ports Collection. In order to do this, run:

```
# rehash
# poudriere ports -c
```

A FreeBSD jail will need to be set up for Poudriere to perform bulk functions. This can be done with the commands:

```
# mkdir /usr/local/poudriere
# poudriere jail -c -j 91x64 -v 12.1-RELEASE -a amd64
```

In the above command, `91x64` identifies the jail, while `12.1-RELEASE` identifies the FreeBSD version to use. These can be tweaked if a different name or FreeBSD version is required. Just remember to also replace them in the rest of this guide.



```
=====line 4 col 13 lines from top 4 =======
www/firefox
x11-wm/i3
irc/irssi
sysutils/tmux
```

Step 4: Creating a Bulk List of Ports

Before continuing, run the following:

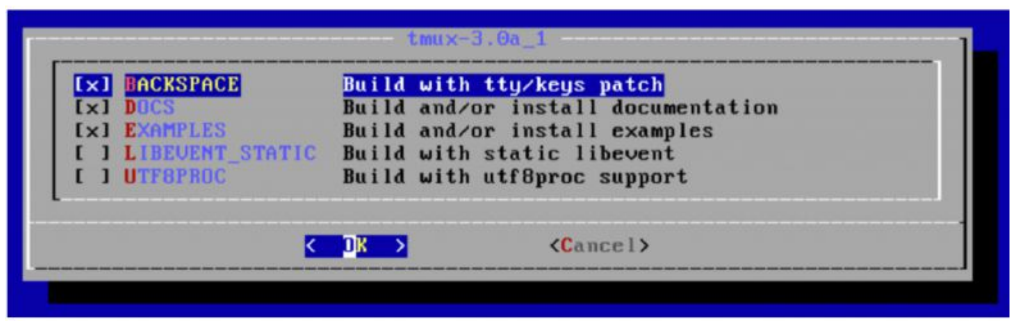
```
# cd poudriere.d
# echo WITH_PKGNG=YES >> 91x64-make.conf
```

The next step will be to create a list of ports for Poudriere to compile and maintain; a wide range of ports is available in the FreeBSD Ports Collection, and Poudriere can be used to manage them all. Start with:

```
# cd /usr/local/etc
# ee poudriere-list
```

As before, use the `ee(1)` text editor to edit the file, adding a list of ports for Poudriere to manage. Above is an example image including Firefox, the i3 Windows manager, irssi, and tmux. Ports can be added to the list by using the port origin (category/name).

Type `ESCAPE` and then `ENTER` twice to exit and save changes to the configuration file.



```
tmux-3.0a_1
[x] BACKSPACE        Build with tty/keys patch
[x] DOCS             Build and/or install documentation
[x] EXAMPLES         Build and/or install examples
[ ] LIBEVENT_STATIC  Build with static libevent
[ ] UTF8PROC         Build with utf8proc support
```

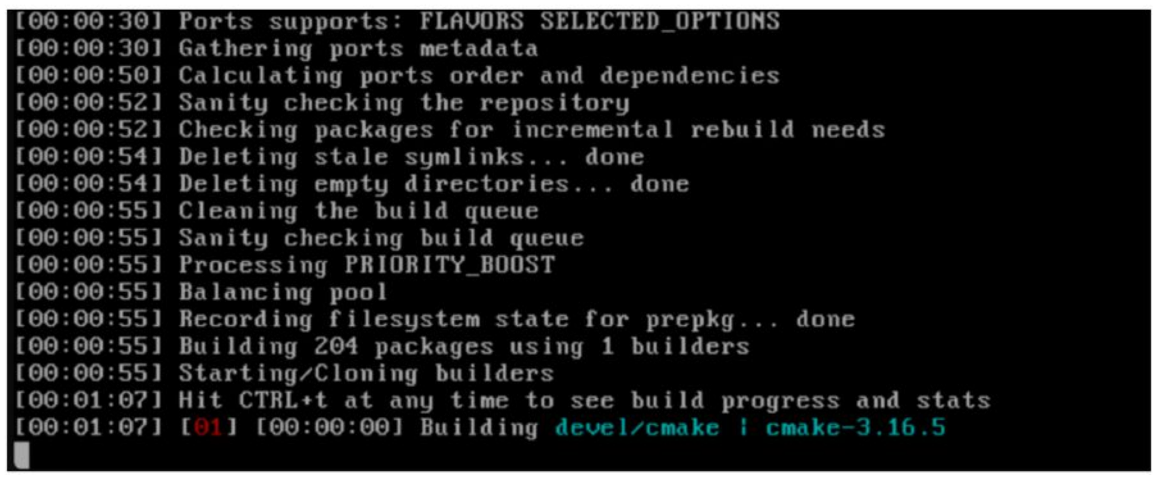
Step 5: Configuring Poudriere Installation Options

This step is optional unless manual configuration is needed.

While Poudriere can be used to automate bulk port management, it still allows users to manually configure each port. However, this step can be done before installation instead of having to be present for the entirety of the process. If, for instance, the user wants to edit tmux(1), use the command:

```
# poudriere options  -c sysutils/tmux
```

Then, using the installation prompts (arrows keys to navigate, space to select) the port can be manually edited. This process can be done for each port that Poudriere manages.



```
[00:00:30] Ports supports: FLAVORS SELECTED_OPTIONS
[00:00:30] Gathering ports metadata
[00:00:50] Calculating ports order and dependencies
[00:00:52] Sanity checking the repository
[00:00:52] Checking packages for incremental rebuild needs
[00:00:54] Deleting stale symlinks... done
[00:00:54] Deleting empty directories... done
[00:00:55] Cleaning the build queue
[00:00:55] Sanity checking build queue
[00:00:55] Processing PRIORITY_BOOST
[00:00:55] Balancing pool
[00:00:55] Recording filesystem state for prepkg... done
[00:00:55] Building 204 packages using 1 builders
[00:00:55] Starting/Cloning builders
[00:01:07] Hit CTRL+t at any time to see build progress and stats
[00:01:07] [01] [00:00:00] Building devel/cmake | cmake-3.16.5
```

Step 6: Using Poudriere to Manage Ports

With configuration complete, installing the entire list of ports can be done with just one command:

```
# poudriere bulk -j 91x64 -f poudriere-list
```

Poudriere will take some time to complete the bulk process, but unlike manually building and installing ports, it can do all the work without needing user inputs. Poudriere also offers a text-based visual display of the installation process. It will allow users to have greater control over their ports without sacrificing automation.

If an update is needed to the port list, Poudriere can update the Ports Collection with the commands:

```
# cd /usr/local/etc
# poudriere bulk -j 91x64 -f poudriere-list
```

DREW GURKOWSKI, FreeBSD Foundation