CapBUG Jan 2013
FreeBSD Jails

Plus OpenBSD ChrootDirectory for building Ports
Michael Erdely <mike@erdelynet.com>
About Jails

- Uses chroot(8) and chroot(2) to change root directory of process and its children
- Uses jail(8) to launch processes in jail
- Unprivileged processes on host can work with privileged ones in jail to gain root on host.
Four Elements of Jails

1. Directory subtree
2. Hostname
3. IP Address
4. Process to run in jail
Jail Types

1. Complete jails are like whole FreeBSD systems
2. Service jails only run one process or service
Creating a Jail

1. export DIR=/path/to/jail
2. mkdir -p ${DIR}
3. cd /usr/src
4. make buildworld
5. make installworld DESTDIR=${DIR}
6. make distribution DESTDIR=${DIR}
7. mount -t devfs devfs ${DIR}/dev

Uses Bourne Shell for setting variable
Configuring the Jail

(Note: 'make distribution' installs etc)

- rc.conf contains:
  - jail_enable="YES"
  - jail_list="jail1 jail2"
  - jail_jail1_rootdir="/path/to/jail1"
  - jail_jail1_hostname="jail1.capbug.org"
  - jail_jail1_ip="10.20.30.40"
  - jail_jail1_devfs_enable="YES"
  - jail_jail1_devfs_ruleset="jail1_ruleset"
Starting a Jail

- Start jail: service jail jail1 start
- Stop jail: service jail jail1 stop
  - Inside jail: sh /etc/rc.shutdown
  - Outside, run rc.shutdown with jexec(8)
Interacting with Jails

- `jls(8)` prints list of jails with JID
- `jexec(8)` launches processes in jail: `jexec 2 bash`

- For service jails, `jail_jailname_exec_start` in `rc.conf(5)` should specify the process to launch
Tuning Jails

- Several sysctl(8) variables are used to configure jail settings in the kernel: security.jail.*
- jls(8) and jexec(8) are part of base
- Other tools in Ports: sysutils/jailutils
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>jps</td>
<td>List processes in a jail</td>
</tr>
<tr>
<td>jid</td>
<td>Print id of a jail</td>
</tr>
<tr>
<td>jstart</td>
<td>Start up a jail securely</td>
</tr>
<tr>
<td>jkill</td>
<td>Shutdown jail in an orderly fashion</td>
</tr>
<tr>
<td>jails</td>
<td>List running jails</td>
</tr>
<tr>
<td>injail</td>
<td>Determine if process is in a jail</td>
</tr>
</tbody>
</table>

http://memberwebs.com/nielsen/freebsd/jails/jailutils/
ezjail Package Description

This port contains two scripts to easily create, manipulate and run FreeBSD jails.

WWW:
http://erdgeist.org/arts/software/ezjail/
Credits

All that I learned about Jails I learned from the FreeBSD Handbook:

OpenSSH ChrootDirectory

For building ports in OpenBSD without littering your main system
Utilize Existing System

- My laptop is powerful:
  - 4 cores, 8 GB RAM, SSD
- I don't want to litter my system with unneeded build dependencies
- Set up "copy" of system in another directory to build ports in
- Use ChrootDirectory in sshd_config to lock ports build user in "jail"
Set up Ports Build Root

1. Pick chroot directory: /home/chroot
2. Extract sets but *etc* in chroot:
   a. `tar -C /home/chroot \`
      -xvzphf <set>.tgz
3. Extract var from etc:
   a. `tar -C /home/chroot \`
      -xvzphf etc*.tgz ./var
4. Copy live /etc to chroot
Set up Ports Build Root Con't

- Make /dev in chroot:
  - cd /home/chroot/dev
  - sh ./MAKEDEV all

- Filesystem complications:
  - /home/chroot can't have nodev and nosuid
Filesystem Layout

- /dev/sd0l on /home/chroot type ffs (local)
- /dev/sd0m on /home/chroot/usr/ports type ffs (local, nodev, nosuid)
- /dev/sd0n on /home/chroot/usr/src type ffs (local, nodev, nosuid)
- /dev/sd0p on /home/chroot/home type ffs (local, nodev, nosuid)
- /dev/sd0f on /home/chroot/usr/obj type ffs (local, nodev, nosuid)
Links into Chroot

- So I can have access to ports in main system, I have:
  - `ln -s /home/chroot/usr/ports /usr/ports`
  - `ln -s /home/chroot/usr/src /usr/src`
I created a separate user ("mwe") on host system (and in chroot)
Edit /etc/ssh/sshd_config (on host):
  Match User mwe
  ChrootDirectory /home/chroot
Restart sshd on host
Build ports

- From host system, connect to chroot:
  - ssh mwe@localhost
- Build the ports
- On host system, install packages from chroot's /usr/ports/packages