FreeBSD in the cloud

FreeBSD & ZFS VPS from a user's perspective

Who's talking?

- Unix admin since 1987
- FreeBSD user since 1.1.5.1
- Committer, 1999-2003
- KFU.COM - ISP 1993-2001 (or so), now personal domain.

What prompted this?

- Servers require static IP addresses.
- Consumer grade internet service with static addressing is limited and costly
- Speed envy was the last straw
  - 50 mbps >> 6 mbps (but really 10 mbps >> 768 kbps)

Phase 1: Conversion to ZFS

- Before VPS, I set up a mirrored ZFS tank config after yet another disk failure.
- ZFS as the root can be done, but is an intensely manual procedure. See URLs at the end for procedure.
- Very important to not skip any steps. One in particular is the zpool.cache file.

ZFS gotchas

- ZFS requires a lot of kernel memory.
  - Remember this for later when we talk about VPS.
- ZFS does not have good recovery tools
  - At the same time, it (appears to) not need them as much.

ZFS benefits

- Snapshots that are fast and *work*
- Granular “filesystem” creation
  - Filesystems are like cleenex
  - Filesystems share common pool of free space
    - Reservations can be used to insure availability

Snapshot strategy

- Backups serve two purposes
  - “Oops. I didn’t mean to remove that file.”
  - “Oops. The disk just exploded.”

Snapshot strategy: backup

- Took my cue from Time Machine
- Hourly snapshots that last a day
- Daily snapshots that last a week
- Weekly snapshots that last a month
- zfsnap

Snapshot strategy: DR

- DR = Disaster Recovery
- Send snapshots offsite: ‘zfs send’
- I wrote a script that is in /usr/local/periodic/weekly
- Use bzip2 and ssh to send the latest weekly snapshot offsite
- Use ssh with a special public key that has a fixed command - “shotput.pl”

Filesystem specialization

- Not everything needs to be backed up.
- Back up the daily postgresql backup, not the actual database.
- Back up the Cyrus IMAP mail partition, but don’t bother backing up the partition metadata
  - metapartition configuration directive
- Turn off atime on root, /usr

Filesystem specialization

- Don’t snapshot /usr/src, /usr/obj or /usr/ports at all

Swap on ZFS

- zfs create -V - create a virtual block device - creates node in /dev/zvol/
- zfs set org.freebsd:swap=on
- zfs_enable=”YES” in rc.conf: enables checking for swap devices.
- Not a panacea: heavy swap usage (like building a JDK port) causes livelocks if you have insufficient kernel memory

Phase 2: To the cloud!

- I picked rootbsd.net
- Advantages:
  - FreeBSD is actually supported!
  - You can ask them to provision your machine without setup, and with the install DVD mounted for booting to perform a custom installation
- Disadvantages:
  - Stingy with RAM

quack.kfu.com

- I chose their Omicron offering
  - 768M RAM
    - This is *awfully* tight for a ZFS config
  - 40 GB disk
  - 500 GB/mo I/O
  - 10 GB of backup disk

ZFS tuning

- /boot/loader.conf:
  - vm.kmem_size="330M"
  - vm.kmem_size_max="330M"
  - vfs.zfs.arc_max="40M"
  - vfs.zfs.vdev.cache.size="5M"
  - vfs.zfs.prefetch_disable=1

Ok, you paid... now what?

- Your console is a VNC server.
  - You could configure it for X, in principle, but XVnc is a better choice
- Since you're going to do root on ZFS, you're going to boot up the Live DVD and do a manual install with the 'fixit'.
  - You can open support tickets to ask them to mount the DVD for you whenever (but there's a time lag).
- Power-cycle the VPS and the DVD will go away.

Filesystem layout

- tank (mountpoint=legacy)
  - tank/usr
    - tank/usr/src
    - tank/usr/obj
    - tank/usr/ports
  - tank/var
  - tank/home
    - tank/home/pgsql, tank/home/imap-spool, tank/home/imap-meta

And now... Xen

- Xen is how our VPS is provisioned.
- AMD64 is recommended arch.
  - /sys/amd64/conf/XENHVM will build a kernel designed to interface directly with Xen
    - One caveat: the kernel will panic without a patch
      - xn0 panic: “do something smart”

Speaking of panic...

- What if you create an unbootable kernel?
  - Oh, go into the loader, unload, load /boot/kernel.old/kernel, etc, boot -s
- No! zpool.cache loading is magical. There is no good solution for this at present.
  - The best I have found is to unload, set the kernel path variable to /boot/kernel.old/ and then boot -s and have it load everything. Got this to work once...

Kernel tuning

- DEVICE_POLLING
- SW_WATCHDOG
  - watchdogd_enable=”YES” in rc.conf
- NO_ADAPTIVE_{MUTEXES,RWLOCKS,SX}
  - kern.hz=100 (in /boot/loader.conf)

Operational suggestions

- ssh rumpelstiltskin attacks
  - for $diety's sake, use keys and turn off passwords!
    - ChallengeResponseAuthentication no
  - bruteblockd to turn logging volume down
    - 4 botched auth attempts -> 10 minute "time out"

Operational suggestions

- Watch out for bandwidth spikes
  - Strategic dummynet application - HTTP capped at 1 MB/sec to avoid potential overage charges
- disable root pw, use sudo

Xenstore Exploration

- We're living in a Xen domU. We know that.
- We can examine our (small) world.
- The xenstore is our window.
- I owe the community a xen-client port. Sorry.
- xenstore-ls device
- xenstore-read /local/domain/0/backend/vif/192/0/mac

Xenstore future

- Actually communicate with the VPS provider?
- A real-time network odometer?
  - /usr/local/periodic script to warn of impending overage?

Other ideas

- An interface between Xen and watchdog(9)?
- Have dom0 power-cycle us if the watchdog timer expires

To-do list

- Commit the “do something smart” patch and merge back
- Fix swap-on-ZFS livelocking
  - Also nice if livelocks would trigger watchdog(9) somehow
- De-magic-ify zpool.cache loading support in loader
  - In other words, make it easier to boot an alternate kernel
- Support root on ZFS in the installer

References:

- Ports:
  - sysutils/zfsnap
  - security/bruteblock
  - security/sudo

References

- PRs:
  - kern/154302: xn0 panic: “do something smart”
  - kern/153804: zpool.cache loading is too magical

References

- URLs:
  - http://www.rootbsd.net/
  - (tbd: my ZFS send / shotput scripts)
  - http://wiki.freebsd.org/RootOnZFS/GPTZFSBoot/Mirror
  - http://wiki.freebsd.org/ZFSTuningGuide

Questions?