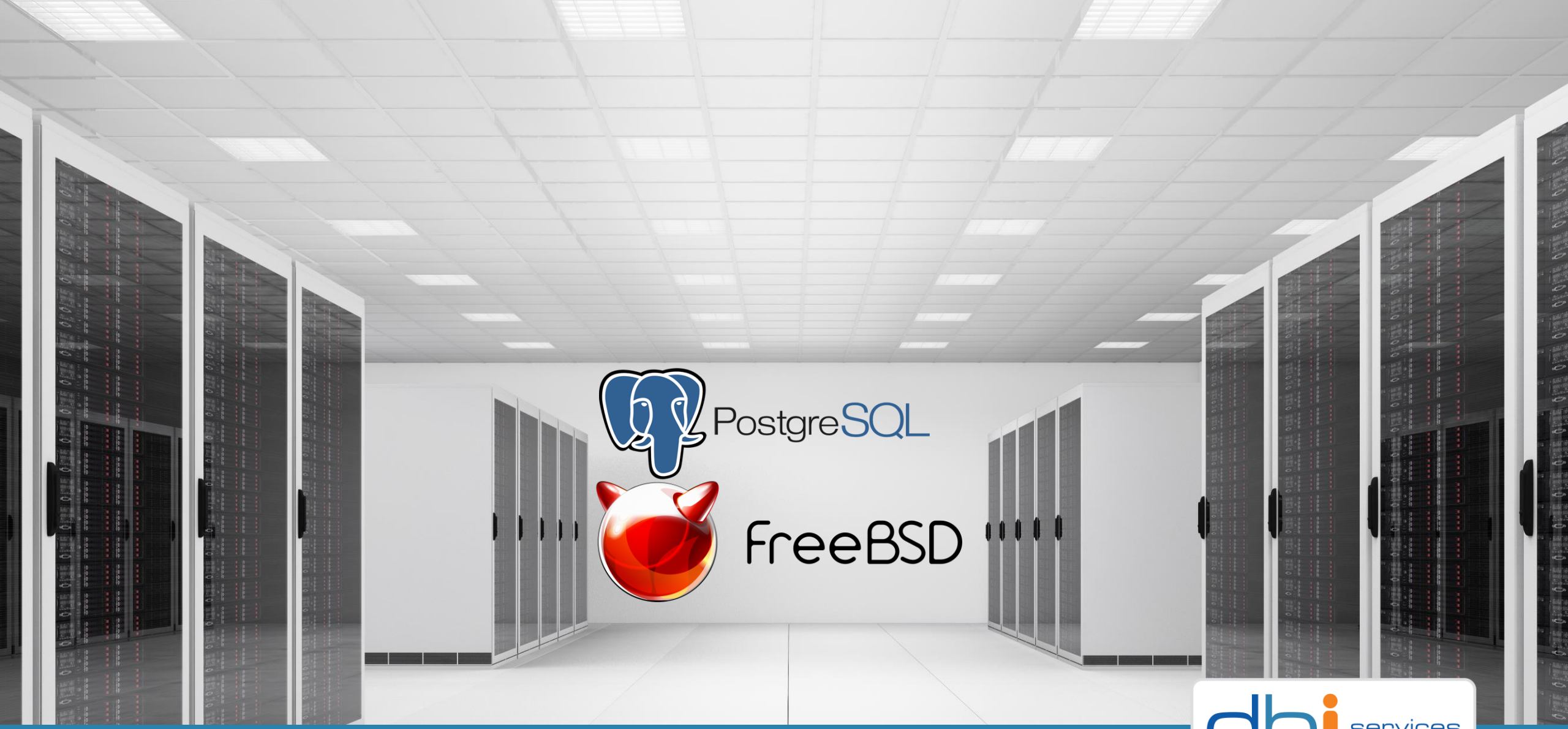


Riding the daemon - PostgreSQL on FreeBSD

dbi services by Sequotech

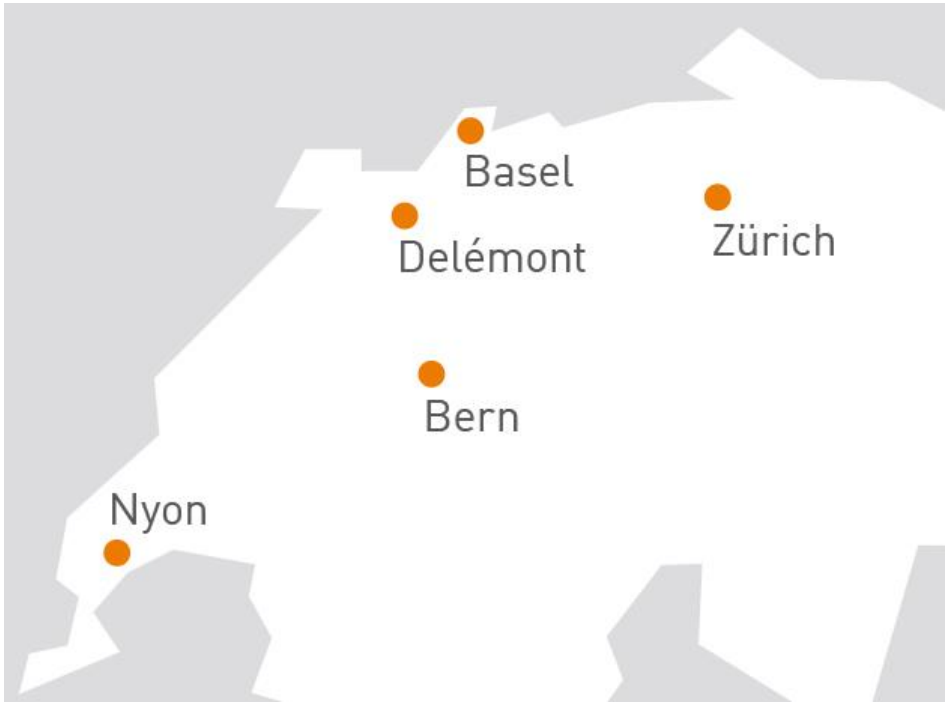
Who we are

The Company

- Founded in 2010
- More than 100 employees
- Specialized in the Middleware Infrastructure
  - The invisible part of IT
- Customers in Switzerland and all over Europe

Our Offer

- Consulting
- Service Level Agreements (SLA)
- Trainings
- License Management



swiss safety center ISO 20000 certified system

swiss safety center ISO 27001 certified system

About me

Daniel Westermann

Technology Leader

Principal Consultant

+41 79 927 24 46

daniel.westermann@dbi-services.com

Daniel Westermann

https://mastodon.social/@danielwestermann


No agenda … Just demo ….

Startup
We´ll start from a minimal base installation

```
root@freebsd14-latest:~ # cat /etc/os-release
NAME=FreeBSD
VERSION="14.2-RELEASE-p1"
VERSION_ID="14.2"
ID=freebsd
ANSI_COLOR="0;31"
PRETTY_NAME="FreeBSD 14.2-RELEASE-p1"
CPE_NAME="cpe:/o:freebsd:freebsd:14.2"
HOME_URL="https://FreeBSD.org/"
BUG_REPORT_URL=https://bugs.FreeBSD.org/

root@freebsd14-latest:~ # which bash
root@freebsd14-latest:~ # echo $SHELL
/bin/sh
```

> Bash is not available by default (or whatever Shell you prefer), the default is "sh"

Startup

This directly brings us to pkg

The package manager on FreeBSD is pkg

```
root@freebsd14-latest:~ # pkg search bash
bash-5.2.37                    GNU Project's Bourne Again SHell
bash-completion-2.14.0,2       Programmable completion library for Bash
bash-completion-freebsd-1.3.0  FreeBSD-oriented extension to the Bash completion library
bash-static-5.2.37             GNU Project's Bourne Again SHell
bashc-5.1.8_1                  GNU Bourne Again shell extended with two-panel file manager
bashdb-5.0.1.1.2_1             Source-code debugger for bash
bashtop-0.9.25_1               Linux/OSX/FreeBSD resource monitor
checkbashisms-2.24.1           Check for the presence of bashisms
hs-dhall-bash-1.0.41           Compile Dhall expressions to bash
imgurbash2-3.3                 Upload or delete images and videos to imgur
mybashburn-1.0.2_6             Ncurses CD burning bash script
p5-Bash-Completion-0.008_2     Extensible system to provide bash completion
p5-Term-Bash-Completion-Generator-0.02.8_2 Generate bash completion scripts
switchBashZsh-1.3_1            Portable shell setup for Bash/Zsh across FreeBSD/Linux/Cygwin
```

Startup
This directly bring us to pkg

Installing Bash

```
root@freebsd14-latest:~ # pkg install bash
Updating FreeBSD repository catalogue...
FreeBSD repository is up to date.
All repositories are up to date.
The following 4 package(s) will be affected (of 0 checked):

New packages to be INSTALLED:
        bash: 5.2.37
        gettext-runtime: 0.23
        indexinfo: 0.3.1
        readline: 8.2.13_2

Number of packages to be installed: 4

root@freebsd14-latest:~ # which bash
/usr/local/bin/bash
```

> Binary packages go to /usr/local/bin !

Startup

This directly bring us to pkg

PostgreSQL comes as (a) binary package(s) as well

```
[root@freebsd14-latest ~]# pkg install postgresql17-client-17.4_1 postgresql17-server-17.4_1
postgresql17-contrib-17.4

Updating FreeBSD repository catalogue...
FreeBSD repository is up to date.
All repositories are up to date.
The following 14 package(s) will be affected (of 0 checked):

New packages to be INSTALLED:
        icu: 74.2_1,1
        libedit: 3.1.20240808,1
        libffi: 3.4.6
        liblz4: 1.10.0,1
…
```

Startup

This directly bring us to pkg

Binary packages can give a lot of informational messages

```
etc...
=====
Message from postgresql17-server-17.2:

--
For procedural languages and postgresql functions, please note that
you might have to update them when updating the server.

If you have many tables and many clients running, consider raising
kern.maxfiles using sysctl(8), or reconfigure your kernel
appropriately.

The port is set up to use autovacuum for new databases, but you might
also want to vacuum and perhaps backup your database regularly. There
is a periodic script, /usr/local/etc/periodic/daily/502.pgsql, that
you may find useful. You can use it to backup and perform vacuum on all
databases nightly. Per default, it performs `vacuum analyze'. See the
script for instructions. For autovacuum settings, please review
~postgres/data/postgresql.conf.
```

Startup
PostgreSQL setup

Initializing a database cluster

```
[root@freebsd14-latest ~]# sysrc postgresql_enable=yes
postgresql_enable:  -> yes
[root@freebsd14-latest ~]# service postgresql initdb
initdb postgresql
The files belonging to this database system will be owned by user "postgres".
This user must also own the server process.

The database cluster will be initialized with this locale configuration:
  locale provider:   libc
  LC_COLLATE:  C
…
Success. You can now start the database server using:

    /usr/local/bin/pg_ctl -D /var/db/postgres/data17 -l logfile start
```

Startup
PostgreSQL setup

You can't install more than one major version when you go with packages

```
[root@freebsd14-latest ~]# pkg install postgresql16-client postgresql16-server postgresql16-contrib
[root@freebsd14-latest ~]# …
Checking integrity... done (0 conflicting)
Conflicts with the existing packages have been found.
One more solver iteration is needed to resolve them.
The following 6 package(s) will be affected (of 0 checked):

Installed packages to be REMOVED:
        postgresql17-client: 17.4_1
        postgresql17-contrib: 17.4
        postgresql17-server: 17.4_1

New packages to be INSTALLED:
        postgresql16-client: 16.8_1
        postgresql16-contrib: 16.8
        postgresql16-server: 16.8_1

Number of packages to be removed: 3
Number of packages to be installed: 3
```

Startup
PostgreSQL setup

You can't install more than one major version when you go with packages

```
[root@freebsd14-latest ~]# which psql
/usr/local/bin/psql
[root@freebsd14-latest ~]# which postgres
/usr/local/bin/postgres
[root@freebsd14-latest ~]# which pg_dump
/usr/local/bin/pg_dump
```

> Everything goes to "/usr/local/bin" without any difference for the major version

Startup
PostgreSQL setup

The FreeBSD init system

```
[root@freebsd14-latest ~]# sysrc -a
dumpdev: AUTO
hostname: freebsd14-latest
ifconfig_vtnet0: DHCP
keymap: ch.kbd
moused_nondefault_enable: NO
postgresql_enable: yes
sshd_enable: YES
zfs_enable: YES

[root@freebsd14-latest ~]# cat /etc/rc.conf
hostname="freebsd14-latest"
keymap="ch.kbd"
ifconfig_vtnet0="DHCP"
sshd_enable="YES"
moused_nondefault_enable="NO"
# Set dumpdev to "AUTO" to enable crash dumps, "NO" to disable
dumpdev="AUTO"
zfs_enable="YES"
postgresql_enable="yes"
```

Startup
PostgreSQL setup

The FreeBSD init system

```
[root@freebsd14-latest ~]# cat /usr/local/etc/rc.d/postgresql
#!/bin/sh

# PROVIDE: postgresql
# REQUIRE: DAEMON mountlate
# KEYWORD: shutdown
#
# Add the following line to /etc/rc.conf to enable PostgreSQL:
#
#  postgresql_enable="YES"
#  # optional
#  postgresql_data="/var/db/postgres/data17"
#  postgresql_flags="-w -s -m fast"
#  postgresql_initdb_flags="--encoding=utf-8 --lc-collate=C"
#  # leave empty to use the login class set in in /etc/passwd:
#  postgresql_login_class="my_custom_login_class"
#  postgresql_profiles=""
```

Startup
PostgreSQL setup

The FreeBSD init system

```
[root@freebsd14-latest ~]# service postgresql start
start postgresql
[root@freebsd14-latest ~]# service postgresql status
status postgresql
pg_ctl: server is running (PID: 1403)
/usr/local/bin/postgres "-D" "/var/db/postgres/data17"
[root@freebsd14-latest ~]# service postgresql xx
/usr/local/etc/rc.d/postgresql: unknown directive 'xx'.
Usage: /usr/local/etc/rc.d/postgresql
[fast|force|one|quiet](start|stop|restart|rcvar|enable|disable|delete|enabled|describe|extra
commands|reload|initdb|status|poll)
[root@freebsd14-latest ~]# su - postgres
$ psql
psql (17.4)
Type "help" for help.

postgres=#
```

Startup
PostgreSQL setup

The default configuration

```
$ psql -c "show data_directory"
     data_directory
-------------------------
 /var/db/postgres/data17
(1 row)

$ psql -c "show hba_file"
              hba_file
-------------------------------------
 /var/db/postgres/data17/pg_hba.conf
(1 row)

$ psql -c "show config_file"
               config_file
-----------------------------------------
 /var/db/postgres/data17/postgresql.conf
(1 row)
```

Startup
PostgreSQL setup

Huge pages on FreeBSD

```
[root@freebsd14-latest ~]# sysctl -a  | grep -i large
vm.pmap.large_map_pml4_entries: 8
largest free range: 524286
vm.largepages.1G: 0
vm.largepages.2M: 0
vm.largepages.reclaim_tries: 1
vfs.zfs.metaslab.df_use_largest_segment: 0
dev.vtnet.0.rx_frame_too_large: 0

[root@freebsd14-latest ~]# psql -c "show shared_memory_size_in_huge_pages"

 shared_memory_size_in_huge_pages
----------------------------------
 -1
(1 row)
```

Startup
PostgreSQL setup

https://www.postgresql.org/docs/17/runtime-config-resource.html

`huge_pages` (`enum`)

Controls whether huge pages are requested for the main shared memory area. Valid values are `try` (the default), `on`, and `off`. With `huge_pages` set to `try`, the server will try to request huge pages, but fall back to the default if that fails. With `on`, failure to request huge pages will prevent the server from starting up. With `off`, huge pages will not be requested. The actual state of huge pages is indicated by the server variable `huge_pages_status`.

At present, this setting is supported only on Linux and Windows. The setting is ignored on other systems when set to `try`. On Linux, it is only supported when `shared_memory_type` is set to `mmap` (the default).

Startup
PostgreSQL setup

The default configuration

```
$ pg_controldata /var/db/postgres/data17/ | grep checksum
Data page checksum version:           0
```

... but:

https://openzfs.github.io/openzfs-docs/Basic%20Concepts/Checksums.html

Checksums and Their Use in ZFS

End-to-end checksums are a key feature of ZFS and an important differentiator for ZFS over other RAID implementations and filesystems. Advantages of end-to-end checksums include:

- detects data corruption upon reading from media
- blocks that are detected as corrupt are automatically repaired if possible, by using the RAID protection in suitably configured pools, or redundant copies (see the zfs `copies` property)
- periodic scrubs can check data to detect and repair latent media degradation (bit rot) and corruption from other sources
- checksums on ZFS replication streams, `zfs send` and `zfs receive`, ensure the data received is not corrupted by intervening storage or transport mechanisms

Startup
PostgreSQL setup

The default configuration

```
$ zfs list
NAME                                          USED  AVAIL  REFER  MOUNTPOINT
zroot                                        3.32G  13.6G    96K  /zroot
zroot/ROOT                                   2.50G  13.6G    96K  none
zroot/ROOT/14.2-RELEASE_2025-02-06_114357       8K  13.6G  1.07G  /
zroot/ROOT/default                           2.50G  13.6G  2.40G  /
zroot/home                                     96K  13.6G    96K  /home
zroot/tmp                                     104K  13.6G   104K  /tmp
zroot/usr                                     831M  13.6G    96K  /usr
zroot/usr/ports                               831M  13.6G   831M  /usr/ports
zroot/usr/src                                  96K  13.6G    96K  /usr/src
zroot/var                                     632K  13.6G    96K  /var
zroot/var/audit                                96K  13.6G    96K  /var/audit
zroot/var/crash                                96K  13.6G    96K  /var/crash
zroot/var/log                                 152K  13.6G   152K  /var/log
zroot/var/mail                                 96K  13.6G    96K  /var/mail
zroot/var/tmp                                  96K  13.6G    96K  /var/tmp
$ zfs get checksum zroot/var
NAME       PROPERTY  VALUE      SOURCE
zroot/var  checksum  on         default
```

Startup
PostgreSQL setup

Creating a new ZFS file system for a new cluster

```
[postgres@freebsd14-latest ~]$ geom disk list
Geom name: vtbd0
Providers:
1. Name: vtbd0
   Mediasize: 21474836480 (20G)
   Sectorsize: 512
   Mode: r2w2e3
rotationrate: unknown
   fwsectors: 63
   fwheads: 16

Geom name: vtbd1
Providers:
1. Name: vtbd1
   Mediasize: 10737418240 (10G)
   Sectorsize: 512
   Mode: r0w0e0
rotationrate: unknown
   fwsectors: 63
   fwheads: 16
```

Startup
PostgreSQL setup

Creating a new ZFS file system for a new cluster

```
root@freebsd14-latest:~ # zpool create pgpool /dev/vtbd1
root@freebsd14-latest:~ # zpool list
NAME     SIZE  ALLOC   FREE  CKPOINT  EXPANDSZ   FRAG    CAP  DEDUP    HEALTH  ALTROOT
pgpool  9.50G   336K  9.50G        -         -     0%     0%  1.00x    ONLINE  -
zroot   17.5G  3.32G  14.2G        -         -     1%    18%  1.00x    ONLINE  -
root@freebsd14-latest:~ # zfs create pgpool/pgzfs
root@freebsd14-latest:~ # zfs set mountpoint=/u02/pgdata/zfs pgpool/pgzfs
root@freebsd14-latest:~ # zfs set compression=lz4 pgpool/pgzfs
root@freebsd14-latest:~ # zfs set recordsize=8K pgpool/pgzfs
root@freebsd14-latest:~ # df -h | grep zfs
pgpool/pgzfs           9.2G     96K    9.2G     0%    /u02/pgdata/zfs
root@freebsd14-latest:~ # chown postgres:postgres /u02/pgdata/zfs
root@freebsd14-latest:~ # su - postgres
```

Startup
PostgreSQL setup

Creating a new cluster

```
[postgres@freebsd14-latest ~]$ /usr/local/bin/initdb --pgdata=/u02/pgdata/zfs
[postgres@freebsd14-latest ~]$ echo "port=5433" >> /u02/pgdata/zfs/postgresql.auto.conf
[postgres@freebsd14-latest ~]$ /usr/local/bin/pg_ctl --pgdata=/u02/pgdata/zfs/ start
[postgres@freebsd14-latest ~]$ /usr/local/bin/psql -p 5433 -l
```

Startup
PostgreSQL setup

Populating both clusters, and comparing the sizes

```
[postgres@freebsd14-latest ~]$ pgbench -i -s 100
[postgres@freebsd14-latest ~]$ pgbench -i -s 100 -p 5433
[postgres@freebsd14-latest ~]$ du -sh /var/db/postgres/data17/
333M    /var/db/postgres/data17/
[postgres@freebsd14-latest ~]$ du -sh /u02/pgdata/zfs/
1.3G    /u02/pgdata/zfs/
root@freebsd14-latest:~ # exit
root@freebsd14-latest:~ # zfs create pgpool/pgzfs2
root@freebsd14-latest:~ # zfs set mountpoint=/u02/pgdata/zfs2 pgpool/pgzfs2
root@freebsd14-latest:~ # chown postgres:postgres /u02/pgdata/zfs2/
root@freebsd14-latest:~ # su - postgres
[postgres@freebsd14-latest ~]$ /usr/local/bin/initdb --pgdata=/u02/pgdata/zfs2
[postgres@freebsd14-latest ~]$ echo "port=5434" >> /u02/pgdata/zfs2/postgresql.auto.conf
[postgres@freebsd14-latest ~]$ /usr/local/bin/pg_ctl --pgdata=/u02/pgdata/zfs2/ start
[postgres@freebsd14-latest ~]$ pgbench -i -s 100 -p 5434
[postgres@freebsd14-latest ~]$ du -sh /u02/pgdata/zfs2/
331M    /u02/pgdata/zfs2/
```

Startup
PostgreSQL setup

Populating both clusters, and comparing the sizes

```
[postgres@freebsd14-latest ~]$ zfs get recordsize /u02/pgdata/zfs/
NAME          PROPERTY    VALUE    SOURCE
pgpool/pgzfs  recordsize  8K       local
[postgres@freebsd14-latest ~]$ zfs get recordsize /u02/pgdata/zfs2/
NAME           PROPERTY    VALUE    SOURCE
pgpool/pgzfs2  recordsize  128K     default
```

Startup
PostgreSQL setup

On a Fedora system

```
postgres@pgbox:/home/postgres/ [pgdev] pgbench -i -s 100
dropping old tables...
creating tables...
generating data (client-side)...
vacuuming...
creating primary keys...
done in 11.89 s (drop tables 0.00 s, create tables 0.02 s, client-side generate 9.09 s,
vacuum 0.15 s, primary keys 2.64 s).
postgres@pgbox:/home/postgres/ [pgdev] du -sh $PGDATA
2.5G    /u02/pgdata/PGDEV/
```

Startup
PostgreSQL setup

The record size matters for performance

```
[postgres@freebsd14-latest ~]$ /usr/local/bin/pgbench --time=15 --port=5433
pgbench (17.4)
starting vacuum...end.
transaction type: <builtin: TPC-B (sort of)>
scaling factor: 100
query mode: simple
number of clients: 1
number of threads: 1
maximum number of tries: 1
duration: 15 s
number of transactions actually processed: 3750
number of failed transactions: 0 (0.000%)
latency average = 4.000 ms
initial connection time = 7.527 ms
tps = 145.338008 (without initial connection time)
```

> https://vadosware.io/post/everything-ive-seen-on-optimizing-postgres-on-zfs-on-linux/

Startup
PostgreSQL setup

The record size matters for performance

```
[postgres@freebsd14-latest ~]$ /usr/local/bin/pgbench --time=15 --port=5434
pgbench (17.4)
starting vacuum...end.
transaction type: <builtin: TPC-B (sort of)>
scaling factor: 100
query mode: simple
number of clients: 1
number of threads: 1
maximum number of tries: 1
duration: 15 s
number of transactions actually processed: 4034
number of failed transactions: 0 (0.000%)
latency average = 3.719 ms
initial connection time = 2.501 ms
tps = 106.812711 (without initial connection time)
```

Startup
PostgreSQL setup

full_page_writes is not required on ZFS

```
[postgres@freebsd14-latest ~]$ psql -p 5433 -c "show full_page_writes"
 full_page_writes
------------------
 on
(1 row)
[postgres@freebsd14-latest ~]$ echo "full_page_writes=off" >>
/u02/pgdata/zfs/postgresql.auto.conf
[postgres@freebsd14-latest ~]$ /usr/local/bin/pg_ctl --pgdata=/u02/pgdata/zfs/ restart
[postgres@freebsd14-latest ~]$ /usr/local/bin/pgbench --time=15 --port=5433
…
latency average = 2.704 ms
initial connection time = 4.844 ms
tps = 369.808903 (without initial connection time)
```

Startup
PostgreSQL setup

You don't need wal_init_zero and wal_recycle on ZFS

```
[postgres@freebsd14-latest ~]$ echo "wal_init_zero=off" >>
/u02/pgdata/zfs/postgresql.auto.conf
[postgres@freebsd14-latest ~]$ echo "wal_recycle=off" >>
/u02/pgdata/zfs/postgresql.auto.conf
[postgres@freebsd14-latest ~]$ /usr/local/bin/pg_ctl --pgdata=/u02/pgdata/zfs/ restart
[postgres@freebsd14-latest ~]$ /usr/local/bin/pgbench --time=15 --port=5433
```

Startup
PostgreSQL setup

Cleaning up & fun with snapshots and clones

```
[postgres@freebsd14-latest ~]$ /usr/local/bin/pg_ctl --pgdata=/u02/pgdata/zfs2/ stop
[postgres@freebsd14-latest ~]$ rm -rf /u02/pgdata/zfs2/*
[postgres@freebsd14-latest ~]$ exit
root@freebsd14-latest:~ # zfs snapshot pgpool/pgzfs@snap1
root@freebsd14-latest:~ # zfs list -t snapshot
NAME                                          USED  AVAIL  REFER  MOUNTPOINT
pgpool/pgzfs@snap1                           8.60M      -  1.41G  -
zroot/ROOT/default@2025-02-06-11:43:57-0       104M      -  1.07G  -
root@freebsd14-latest:~ # zfs create pgpool/clones
root@freebsd14-latest:~ # zfs clone pgpool/pgzfs@snap1 pgpool/clones/1
root@freebsd14-latest:~ # zfs list
…
pgpool/clones                                  104K  7.76G    96K  /pgpool/clones
pgpool/clones/1                                  8K  7.76G  1.41G  /pgpool/clones/1
…
root@freebsd14-latest:~ # ls /pgpool/clones/1/
```

Startup
PostgreSQL setup

Fun with snapshots and clones

```
root@freebsd14-latest:~ # su - postgres
[postgres@freebsd14-latest ~]$ echo "port=5435" >> /pgpool/clones/1/postgresql.auto.conf
[postgres@freebsd14-latest ~]$ rm /pgpool/clones/1/postmaster.pid
[postgres@freebsd14-latest ~]$ /usr/local/bin/pg_ctl --pgdata=/pgpool/clones/1/ start
[postgres@freebsd14-latest ~]$ /usr/local/bin/psql -p 5435 -c "\l+"
 Name     |  Owner   | Encoding | Locale Provider |  Size
----------+----------+----------+-----------------+---------
 postgres  | postgres | UTF8     | libc            | 1507 MB
 template0 | postgres | UTF8     | libc            | 7449 kB
           |          |          |                 |
 template1 | postgres | UTF8     | libc            | 7521 kB
           |          |          |                 |
[postgres@freebsd14-latest ~]$ /usr/local/bin/pg_ctl --pgdata=/pgpool/clones/1/ stop
[postgres@freebsd14-latest ~]$ exit
root@freebsd14-latest:~ # zfs destroy -r pgpool/clones
```

Startup
PostgreSQL setup

Fun with Jails

```
root@freebsd14-latest:~ # sysrc jail_enable="YES"
jail_enable: NO -> YES
root@freebsd14-latest:~ # sysrc jail_parallel_start="YES"
jail_parallel_start: NO -> YES
root@freebsd14-latest:~ # zfs create -o mountpoint=/usr/local/jails zroot/jails
root@freebsd14-latest:~ # zfs create zroot/jails/media
root@freebsd14-latest:~ # zfs create zroot/jails/templates
root@freebsd14-latest:~ # zfs create zroot/jails/containers
root@freebsd14-latest:~ # fetch https://download.freebsd.org/ftp/releases/amd64/amd64/14.2-
RELEASE/base.txz -o /usr/local/jails/media/14.2-RELEASE-base.txz
root@freebsd14-latest:~ # tar -xf /usr/local/jails/media/14.2-RELEASE-base.txz -C
/usr/local/jails/templates/14.2-RELEASE --unlink
oot@freebsd14-latest:~ # ls -l /usr/local/jails/templates/14.2-RELEASE
total 94
-rw-r--r--   2 root wheel 1011 Nov 29 10:58 .cshrc
-rw-r--r--   2 root wheel  495 Nov 29 10:58 .profile
-r--r--r--   1 root wheel 6109 Nov 29 11:42 COPYRIGHT
drwxr-xr-x   2 root wheel   49 Nov 29 10:58 bin
…
```

Startup
PostgreSQL setup

Fun with Jails

```
root@freebsd14-latest:~ # cp /etc/resolv.conf /usr/local/jails/templates/14.2-
RELEASE/etc/resolv.conf
root@freebsd14-latest:~ # cp /etc/localtime /usr/local/jails/templates/14.2-
RELEASE/etc/localtime
root@freebsd14-latest:~ # freebsd-update -b /usr/local/jails/templates/14.2-RELEASE/ fetch
install
root@freebsd14-latest:~ # zfs snapshot zroot/jails/templates/14.2-RELEASE@base
root@freebsd14-latest:~ # zfs list -t snapshot
NAME                                          USED  AVAIL  REFER  MOUNTPOINT
pgpool/pgzfs@snap1                           32.9M      -  1.41G  -
zroot/ROOT/default@2025-02-06-11:43:57-0       104M      -  1.07G  -
zroot/jails/templates/14.2-RELEASE@base          0B      -   450M  -
root@freebsd14-latest:~ # zfs clone zroot/jails/templates/14.2-RELEASE@base
zroot/jails/containers/thinjail1
root@freebsd14-latest:~ # ls -l /usr/local/jails/containers/thinjail1/
total 94
-rw-r--r--   2 root wheel 1011 Nov 29 10:58 .cshrc
-rw-r--r--   2 root wheel  495 Nov 29 10:58 .profile
…
```

Startup
PostgreSQL setup

Fun with Jails, thin Jails are pretty small

```
root@freebsd14-latest:~ # df -h | grep jails
zroot/jails                          13G     96K     13G     0%    /usr/local/jails
zroot/jails/containers               13G     96K     13G     0%
/usr/local/jails/containers
zroot/jails/templates                13G     96K     13G     0%
/usr/local/jails/templates
zroot/jails/media                    13G    197M     13G     1%    /usr/local/jails/media
zroot/jails/templates/14.2-RELEASE   13G    450M     13G     3%
/usr/local/jails/templates/14.2-RELEASE
zroot/jails/containers/thinjail1     13G    450M     13G     3%
/usr/local/jails/containers/thinjail1
```

Startup
PostgreSQL setup

Fun with Jails, the Jail configuration

```
root@freebsd14-latest:~ # cat /etc/jail.conf.d/thinjail1.conf
thinjail1 {
  # STARTUP/LOGGING
  exec.start = "/bin/sh /etc/rc";
  exec.stop = "/bin/sh /etc/rc.shutdown";
  exec.consolelog = "/var/log/jail_console_${name}.log";

  # PERMISSIONS
  allow.raw_sockets;
  exec.clean;
  mount.devfs;

  # HOSTNAME/PATH
  host.hostname = "${name}";
  path = "/usr/local/jails/containers/${name}";

  # NETWORK
  ip4.addr = 192.168.122.160;
  interface = vtnet0;
}
```

Startup
PostgreSQL setup

Fun with Jails, working with Jails

```
root@freebsd14-latest:~ # service jail start thinjail1
Starting jails: thinjail1.
root@freebsd14-latest:~ # service jail status thinjail1
 JID             IP Address      Hostname            Path
 thinjail1       192.168.122.160 thinjail1           /usr/local/jails/containers/thinjail1

root@freebsd14-latest:~ # jexec -u root thinjail1
root@thinjail1:/ # ifconfig
vtnet0: flags=1008843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST,LOWER_UP> metric 0 mtu 1500

options=4c07bb<RXCSUM,TXCSUM,VLAN_MTU,VLAN_HWTAGGING,JUMBO_MTU,VLAN_HWCSUM,TSO4,TSO6,LRO,VLA
N_HWTSO,LINKSTATE,TXCSUM_IPV6>
        ether 52:54:00:35:84:81
        inet 192.168.122.160 netmask 0xffffffff broadcast 192.168.122.160
        media: Ethernet autoselect (10Gbase-T <full-duplex>)
        status: active
lo0: flags=1008049<UP,LOOPBACK,RUNNING,MULTICAST,LOWER_UP> metric 0 mtu 16384
        options=680003<RXCSUM,TXCSUM,LINKSTATE,RXCSUM_IPV6,TXCSUM_IPV6>
        groups: lo
```

Startup
PostgreSQL setup

Fun with Jails, adding PostgreSQL to the Jail

```
root@freebsd14-latest:~ # service jail stop thinjail1
Stopping jails: thinjail1.
root@freebsd14-latest:~ # chflags -R 0 /usr/local/jails/containers/thinjail1
root@freebsd14-latest:~ # rm -rf /usr/local/jails/containers/thinjail1/*
root@freebsd14-latest:~ # zfs destroy zroot/jails/containers/thinjail1
root@freebsd14-latest:~ # jls
   JID  IP Address      Hostname                      Path
root@freebsd14-latest:~ # zfs destroy zroot/jails/templates/14.2-RELEASE@base
```

Startup
PostgreSQL setup

Fun with Jails, adding PostgreSQL to the Jail

```
root@freebsd14-latest:~ # pkg -c /usr/local/jails/templates/14.2-RELEASE/ install
postgresql17-client-17.4_1 postgresql17-server-17.4_1 postgresql17-contrib-17.4
root@freebsd14-latest:~ # sysrc -R /usr/local/jails/templates/14.2-RELEASE/
postgresql_enable=yes
postgresql_enable:  -> yes
root@freebsd14-latest:~ # zfs snapshot zroot/jails/templates/14.2-RELEASE@postgresql17
```

Startup
PostgreSQL setup

Fun with Jails, adding PostgreSQL to the Jail

```
root@freebsd14-latest:~ # cat /var/tmp/create_jail.sh
#!/usr/local/bin/bash
# The parameters for a new Jail
JAILNAME="$1"
IPADDR="$2"
# This is where all the Jails go to
BASEPATH="/usr/local/jails/containers/"
cat << EOF > /etc/jail.conf.d/${JAILNAME}.conf
${JAILNAME} {
  # STARTUP/LOGGING
  exec.start = "/bin/sh /etc/rc";
  exec.stop = "/bin/sh /etc/rc.shutdown";
  exec.consolelog = "/var/log/jail_console_${name}.log";
  # PERMISSIONS
  allow.raw_sockets;
  allow.sysvipc = 1;
  exec.clean;
  mount.devfs;
  # HOSTNAME/PATH
  host.hostname = "${JAILNAME}.it.dbi-services.com";
  path = "${BASEPATH}/${JAILNAME}";
  # NETWORK
  ip4.addr = ${IPADDR};
  interface = vtnet0;
}
EOF
zfs clone zroot/jails/templates/14.2-RELEASE@postgresql17 zroot/jails/containers/${JAILNAME}
service jail start ${JAILNAME}
jls
jexec -U postgres ${JAILNAME} /usr/local/bin/initdb --pgdata=/var/db/postgres/data17
jexec -l ${JAILNAME} service postgresql start
```

Startup
PostgreSQL setup

Fun with Jails, adding PostgreSQL to the Jail

```
root@freebsd14-latest:~ # chmod +x /var/tmp/create_jail.sh
root@freebsd14-latest:~ # /var/tmp/create_jail.sh PG1 192.168.122.130
root@freebsd14-latest:~ # psql -h 192.168.122.130
psql: error: connection to server at "192.168.122.130", port 5432 failed: FATAL:  no
pg_hba.conf entry for host "192.168.122.130", user "root", database "root", no encryption
root@freebsd14-latest:~ # du -sh /usr/local/jails/containers/PG1
1.8G    /usr/local/jails/containers/PG1
```

Startup
PostgreSQL setup

Fun with Linux containers

```
root@freebsd14-latest:~ # pkg install podman
Updating FreeBSD repository catalogue...
FreeBSD repository is up to date.
All repositories are up to date.
The following 26 package(s) will be affected (of 0 checked):

New packages to be INSTALLED:
        conmon: 2.1.12_4
        containernetworking-plugins: 0.5_10
        containers-common: 0.60.4_1
root@freebsd14-latest:~ # sysrc podman_enable=YES
podman_enable:  -> YES
root@freebsd14-latest:~ # sysrc linux_enable=YES
linux_enable: NO -> YES
root@freebsd14-latest:~ # service linux start
root@freebsd14-latest:~ # mount -t fdescfs fdesc /dev/fd
```

Startup
PostgreSQL setup

Fun with Linux containers

```
root@freebsd14-latest:~ # echo "unqualified-search-registries = ['docker.io']" >>
/usr/local/etc/containers/registries.conf
root@freebsd14-latest:~ # podman search postgres
NAME                                DESCRIPTION
docker.io/library/postgres          The PostgreSQL object-relational database sy...
docker.io/circleci/postgres         The PostgreSQL object-relational database sy...
```

Startup
PostgreSQL setup

Fun with Linux containers, configure networking

```
root@freebsd14-latest:~ # cp /usr/local/etc/containers/pf.conf.sample /etc/pf.conf
root@freebsd14-latest:~ # sysrc pf_enable=YES
pf_enable: NO -> YES
root@freebsd14-latest:~ # service pf start
Enabling pf.
root@freebsd14-latest:~ # sysctl net.pf.filter_local=1
net.pf.filter_local: 0 -> 1
root@freebsd14-latest:~ # service pf restart
Disabling pf.
Enabling pf.
```

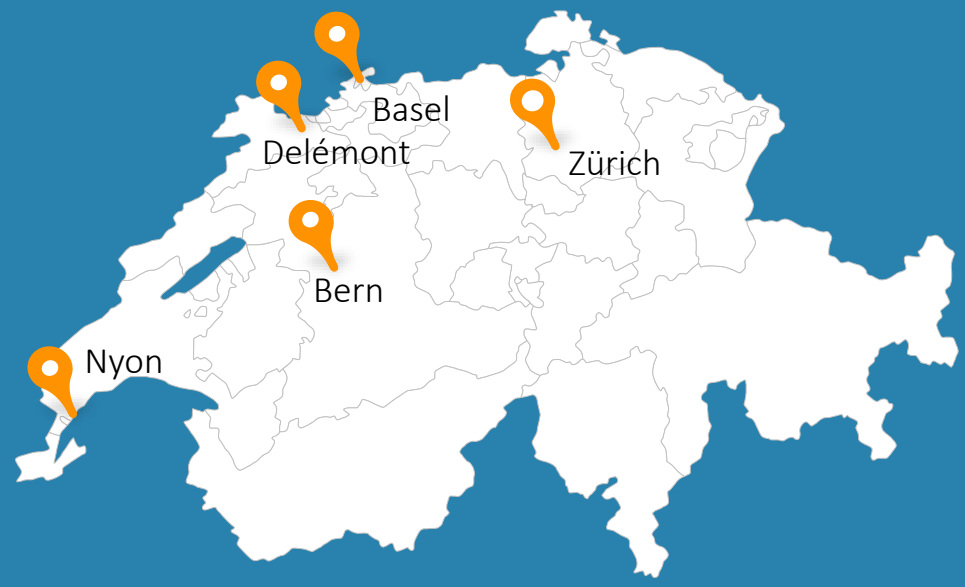
Startup
PostgreSQL setup

Fun with Linux containers, running a container

```
root@freebsd14-latest:~ # podman search alpine
NAME                           DESCRIPTION
docker.io/library/alpine       A minimal Docker image based on Alpine Linux...
docker.io/alpine/git           A  simple git container running in alpine li...
docker.io/alpine/socat         Run socat command in alpine container
…
root@freebsd14-latest:~ # podman run -it --rm --replace --name my-postgres --platform
linux/amd64 docker.io/library/alpine /bin/sh
/ # cat /etc/os-release
NAME="Alpine Linux"
ID=alpine
VERSION_ID=3.21.3
PRETTY_NAME="Alpine Linux v3.21"
HOME_URL="https://alpinelinux.org/"
BUG_REPORT_URL="https://gitlab.alpinelinux.org/alpine/aports/-/issues"
/ #
```

Any questions?

Please do ask!



We would love to boost your IT-Infrastructure

How about you?