

# POLITECHNIKA WARSZAWSKA

## WYDZIAŁ ELEKTRYCZNY

### Unix - projekt

Piotr Błądek 252 536

12 lutego 2016

## 1 Zadanie

Moje zadanie składało się z dwóch części, pierwsza polegała na instalacji systemu FreeBSD na dysku lokalnym z systemem plików ZFS. Druga natomiast to instalacja portu redmine na przygotowanym wcześniej dysku.

## 2 Wykonanie zadania

Do każdego kroku zadania powstał osobny skrypt. W sumie powstały trzy skrypty, ponieważ krok drugi został podzielony na dwa etapy.

### 2.1 Krok pierwszy - instalacja FreeBSD

W pierwszym kroku ładujemy system *FreeBSD 10.2-STABLE* z obrazu iso. W nim uruchamiamy skrypt który ma za zadanie przygotować dysk do instalacji systemu, a następnie zainstalować sam system. Programem *gpart* tworzę dwie partycje, pierwszą typu *freebsd-boot* na której umieszczam kod bootstrap, drugą typu *freebsd-zfs* na której tworzę zpooła o nazwie *zroot*.

W kolejnych krokach konfiguruje stworzonego wcześniej zpooła tak żebym mógł na nim zainstalować system. Później montuję katalog */ftp/pub* z serwera *amp* i pobieram obraz iso systemu który docelowo znajdzie się na dysku lokalnym - *FreeBSD 10.2* architektura amd64. Montuję go w systemie i rozpakowuję na dysk.

Ostatni etap pierwszego kroku to konfiguracja systemu, ustawiam między innymi że na starcie ma się załadować ZFS, system ma sobie pobrać konfigurację sieciową z DHCP, ma się uruchomić serwis ssh i tym podobne ustawienia startowe.

### 2.2 Krok drugi - instalacja skryptu *base*

Wydzieliłem ten krok z kroku ostatniego. Zadanie w tym kroku polega na załadowaniu systemu z dysku lokalnego i uruchomieniu skryptu który zamontuje zasoby sieciowe i uruchomi

skrypt *base*. Po wykonaniu skryptu można się będzie zalogować na swego użytkownika z LDAP wydziałowego, tak też robimy i przechodzimy do kroku ostatniego.

### 2.3 Krok trzeci - instalacja portu *Redmine*

Ściągamy drzewo portów i rozpakowujemy je na dysku lokalnym. Przed samą instalacją portów podmieniam pliki konfiguracyjne instalacji w katalogu `/var/db/ports`, sam program *make* uruchamiam z podaniem opcji *BATCH=yes* przez co nie prosi on użytkownika o konfigurację podczas instalacji tylko korzysta z ustawień domyślnych, lub jeżeli we wspomnianym wcześniej katalogu znajduje się plik konfiguracyjny to program korzysta właśnie z niego. Zmiana konfiguracji była mi potrzebna żeby zainstalować między innymi moduł *Passenger* do *nginx*, *redmine* oraz *ruby*, przy jej pomocy usunąłem również zależności od pakietu *X11*, *mysql*, *postgresql* i kilku innych które nie są mi potrzebne.

Po instalacji przechodzę do konfiguracji serwisów. Na koniec uruchamiam *nginx*, po tej operacji w przeglądarce pod adresem maszyny (10.146.10.136) powinna nam się ukazać aplikacja *Redmine*.

## 3 Wnioski

Najwięcej nauczyłem się przy kompilacji i instalacji portów. System portów jest w FreeBSD dobrze zaprojektowany, konfiguracja jest prosta i przyjemna. W łatwy sposób można konfigurację sprepować i instalację przeprowadzić automatycznie bez ingerencji użytkownika. W łatwy sposób można wyłączyć zależności od pakietów których nie będziemy używać a które domyślnie są instalowane przy okazji instalacji portów z plików binarnych kompilowanych przez producentów aplikacji.

Przy konfiguracji dysku pod ZFS i instalacji systemu napotkałem kilka problemów które komentuję w samym skrypcie. Nie wiedziałem że instalacja systemu może przebiegać w tak prosty sposób, później konfiguracja wstępna systemu to tylko kilka linii skryptu, które zresztą pisało się bardzo przyjemnie. Pamiętam że na systemie Oracle Linux sama instalacja *Redmine* z plików binarnych zajęła mi kilka godzin, tutaj w te kilka godzin zainstalowałem system, skompilowałem i zainstalowałem *Redmine* i napisałem do tego skryptu żeby robić to automatycznie, wszystko w tym systemie wydawało mi się łatwiejsze i lepiej zaprojektowane niż w RedHatowym linuxie który wybrałem jako serwer dla mojego koła naukowego, gdybym miał wybierać system teraz z pewnością wybrałbym FreeBSD.

## 4 Instrukcja

Po poprawnej instalacji, należy uruchomić port *nginx* `sudo service nginx start`, pod adresem lokalnym maszyny i lokalizacji podanej w pliku konfiguracyjnym *redmine.conf* powinien działać port *redmine*. Do czystego portu można logować się z konta LDAP wydziałowego, takie konto posiada jednak ograniczone prawa (z oczywistych względów). Dlatego należy zalogować się na

użytkownika lokalnego dla portu *admin* z hasłem *admin* , zmienić mu oczywiście hasło i mianować własnego użytkownika administratorem. Zalecam zostawienie użytkownika lokalnego *admin* ponieważ w przypadku problemu z zalogowaniem się przez LDAP, nie będzie możliwości innego dostania się do aplikacji (będzie trzeba ingerować w bazę danych).

W poradniku instalacyjnym redmine, dowiadujemy się że: *SQLite 3 (not for multi-user production use!)*, argumentacja tego stwierdzenia: *That depends of the workload you have and the expected size the redmine will have. Put shortly: you have no caching and no write concurrency as in a zeal"database, the 2 caveats this brings are: no caching means all accesses go directly to the disk, which might cause high I/O load on the disk if you are accessing redmine with many users at the same time, and no concurrency because a write access to the SQLite will lock the entire file, i.e. during the time it takes for data to be written to disk, nothing else can access the DB. Furthermore, even read performance might degrade with the size of the DB file because of high seek times. With all that said: SQLite is not a bad database engine per se, but you have to be aware of the limits of the engine. I'd recommend going with PostgreSQL or MySQL if you will regularly have more than 3-5 concurrent users and/or a lot of issue creation, if not, SQLite should work good enough.* Jednak opisany wcześniej problem będzie dotyczył jedynie bardzo rozbudowanego środowiska, przy małych obciążeniach (i niewielkiej bazie danych) nie powinno być problemów z obsługą portu.

## 5 Załączniki

### 5.1 Instalacja FreeBSD

```
#!/bin/sh
# sys-install
# bladekp 2016

# https://wiki.freebsd.org/RootOnZFS/GPTZFSBoot
# https://forums.freebsd.org/threads/zfs-boot-from-usb.45880/
# https://calomel.org/zfs_freebsd_root_install.html
# https://www.freebsd.org/doc/en/articles/
#       remote-install/preparation.html
# http://louwrentius.com/freebsd-101-unattended-
#       install-over-pxe-http-no-nfs.html
# https://forums.freebsd.org/threads/disk-labels.44612/
# https://forums.freebsd.org/threads/zfs-boot-from-usb.45880/

# volt:/usr/local/zetis/sbin/michtam/*           ZFS-root install
# volt:/usr/local/zetis/sbin/freebsd-devprep
# volt:/usr/local/zetis/sbin/freebsd-install

clean-disk () { # dysk #zroot
    dysk=$1
    zroot=$2
    umount $zroot
    umount /mnt
    zpool destroy $zroot
    gpart delete -i 2 $dysk
    gpart delete -i 1 $dysk
    gpart destroy -F $dysk
}

dev-prep () { # dysk
    dysk=$1
    gpart create -s gpt $dysk
    gpart add -a 4k -s 64k -t freebsd-boot $dysk
    gpart add -a 4k -t freebsd-zfs -l root $dysk
    gpart bootcode -b /boot/pmbr -p /boot/gptzfsboot -i 1 $dysk
    zpool create -f -o altroot=$MPOINT -o
```

```

cachefile=$ZCACHE $ZROOT /dev/gpt/root
  zpool export $ZROOT
  zpool import -o altroot=$MPOINT -o cachefile=$ZCACHE $ZROOT
  zpool set bootfs=$ZROOT $ZROOT
  zfs set compression=lz4 $ZROOT
  zfs set atime=off $ZROOT
}

sys-install () { # root-dir                # punkt montowania
  R=$1
  for file in $SRC/base.txz $SRC/kernel.txz $SRC/doc.txz ; do
    tar --unlink -xpJ -f $file -C $R
  done
  cp $ZCACHE $R/boot/zfs/zpool.cache
  echo 'zfs_enable="YES"' >> $R/etc/rc.conf
  ln -s usr/home $R/home
}

sys-config () { # root-dir                # konfiguracja systemu
  R=$1
  cat >> $R/boot/loader.conf <<EOF
zfs_load="YES"
vfs.root.mountfrom="zfs:$ZROOT"
EOF
  cat >> $R/etc/rc.conf <<EOF
ifconfig_em0=dhcp
sshd_enable=YES
EOF
  touch $R/etc/fstab
}

all () {
  clean-disk $DYSK $ZROOT

  dev-prep $DYSK

  sys-install $MPOINT/$ZROOT

  sys-config $MPOINT/$ZROOT
}

```

```

}

help () {
    echo "uzycie: _sys-install_[-ahc]
    _a_ przygotuj_dysk_i_zainstaluj_freebsd
    _h_ pomoc
    _c_ wyczysc_dysk"
}

# START

VER=10.2-STABLE
VER=10.3-PRERELEASE
ARCH=amd64
ZPOOL=tank
ZROOT=zroot
SRC=/ftp/pub/BSD/FreeBSD/snapshots/$ARCH/$ARCH/$VER
DYSK=da0
MPOINT=/mnt
ZCACHE=/var/tmp/zpool.cache           # lokalizacja kieszeni

case "$1" in
-a) all ;;
-c) clean-disk $DYSK $ZROOT ;;
-h) help ;;
esac

exit 0

```

## 5.2 Instalacja *redmine*

```
#!/bin/sh
# cel: instalacja redmine i konfiguracja w zetis
# bladekp 2016

PORTS=/tmp/ports          # lokalne modyfikacje portow (gorna warstwa)

MYDIR=/ftp/pub/FreeBSD/zetis/bladekp

RUSER=redmine

# 0. srodowisko

export WRKDIRPREFIX=/tmp/obj          # sa w /etc/makefile.conf
export DISTDIR=$PORTS/distfiles #

mkdir -p /ftp/pub
mount amp:/ftp/pub /ftp/pub
ln -s /ftp/pub/FreeBSD/SVN/ports /usr/ports

mkdir $PORTS
/sbin/mount -t unionfs -o ro $PORTS /usr/ports

# 2. Dodanie 2 warstwy plikow dla porty redmine w
# /usr/local/ports/www/redmine # Makefile.local

pdir=$PORTS/www/redmine
mkdir -p $pdir $pdir/files
for x in database.yml secret_token.rb auth_sources.dump.sql
    redmine.conf passenger.conf ; do
    cp $MYDIR/$x $pdir/files
done

cp $MYDIR/Makefile.local $pdir

# 3. Instalacja portow

tar -xf $MYDIR/var-db-ports.tar -C / # rozpakowanie drzewa parametrow
```

```

make -C /usr/ports/www/redmine install BATCH=yes

make -C /usr/ports/www/nginx-devel install clean BATCH=yes

make -C /usr/ports/www/redmine config-zetis clean

exit 0

```

### 5.3 Dodatkowy cel w pliku Makefile.local

```

# bladekp 2016
# cel: konfiguracja redmine na potrzeby zetis

RUSER=www
RGROUP=www

#ADMINPASS=admin123

config-zetis:
    # Kopiowanie konfiguracji.
    cp ${FILESDIR}/passenger.conf ${LOCALBASE}/etc/nginx
    cp ${FILESDIR}/redmine.conf ${LOCALBASE}/etc/nginx
    cp ${FILESDIR}/database.yml ${WWWDIR}/config
    cp ${FILESDIR}/secret_token.rb ${WWWDIR}/config/initializers/
    # Inicjalizacja bazy danych redmine.
    cd ${WWWDIR} ; ${SUDO}
RAILS_ENV=production bundle exec rake db:migrate
    cd ${WWWDIR} ; ${SUDO}
RAILS_ENV=production REDMINE_LANG=pl rake redmine:load_default_data
    # Dodanie konfiguracji logowania z ldap wydzialowego.
    sqlite3 ${WWWDIR}/db/redmine.sqlite3 <
${FILESDIR}/auth_sources.dump.sql
    # Nadanie praw do plikow.
    chown -R ${RUSER}:${RGROUP} ${WWWDIR}/db
    chown -R ${RUSER}:${RGROUP} ${WWWDIR}/tmp
    chown -R ${RUSER}:${RGROUP} ${WWWDIR}/log
    chown -R ${RUSER}:${RGROUP} ${WWWDIR}/files
    chmod -R 0755 ${WWWDIR}
    # Konfiguracja nginx

```



```
{SUDO} perl -pi -e '$$_ .= qq(include passenger.conf; \n
include redmine.conf;\n) if /http {/' ${LOCALBASE}/etc/nginx/nginx.conf
# Dodanie nginx do programow startup.
{SUDO} sysrc -f /etc/rc.conf.local nginx_enable=YES
```

## 5.4 Plik konfiguracyjny nginx *redmine.conf*

```
server {
    listen      80;
    location ~ ^/redmine(/.*|$) {
        alias   /usr/local/www/redmine/public$1;
        passenger_base_uri /redmine;
        passenger_app_root /usr/local/www/redmine;
        passenger_document_root /usr/local/www/redmine/public;
        passenger_enabled on;
    }
}
```