

System Informatyczny

CELAB

Instalacja CELAB na nowym serwerze

Spis treści

1.1. Instalacja serwera z oprogramowaniem CELAB.....	2
1.2. Instalacja oprogramowania systemowego CentOS.....	2
1.3. Konfiguracja bezpiecznego terminala SSH.....	4
1.4. Konfiguracja startowa systemu CentOS.....	5
1.5. Instalacja i konfiguracja serwera bazy danych PostgreSQL.....	7
1.6. Poprawa wydajności serwera baz danych PostgreSQL.....	9
1.7. Instalacja i konfiguracja serwera aplikacyjnego JBoss.....	9
1.8. Wczytanie bazy danych PostgreSQL z pliku archiwalnego backup.....	10
1.9. Instalacja i konfiguracja aplikacji CELAB.....	10

1.1. Instalacja serwera z oprogramowaniem CELAB

Niniejsza instrukcja opisuje procedury, które mogą być wykorzystane przy nowej instalacji serwera przeznaczonego do obsługi systemu CELAB (LIMS i/lub CBD).

W przypadku instalacji nowych serwerów przyjęte zostały następujące założenia:

1. Serwer posiada procesor zgodny z 64-bitową architekturą x86_64.
2. Serwer wyposażony jest w minimum 4GB pamięci RAM.
3. Na serwerze zostanie zainstalowany system operacyjny CentOS w najnowszej dostępnej wersji, charakteryzujący się dużą stabilnością, długim okresem wsparcia i pełną zgodnością z Red Hat Enterprise Linux. System CentOS jest dostępny za darmo.
4. Przed rozpoczęciem instalacji dysponujemy zarchiwizowaną bazą danych programu CELAB (plik stworzony poleceniem `pg_dump -F c`). Baza danych wraz z zawartą w niej licencją jest przygotowywana indywidualnie dla każdej lokalizacji CELAB.

Uwaga!

Instrukcja ta nie dotyczy oryginalnych serwerów CELAB dostarczonych do ZHW wraz z całym systemem w 2006 roku (opartych o system PLD Linux), ponieważ ze względu na postęp technologiczny nie są one aktualnie kompatybilne z najnowszymi rozwiązaniami programowymi. Procedury instalacyjne dotyczące tych serwerów są zawarte w dokumencie 3.01. Instalacja serwera.

1.2. Instalacja oprogramowania systemowego CentOS

1. Oprogramowanie powinno być instalowane na architekturze 64 bitowej. W związku z powyższym należy przygotować płytę instalacyjną DVD systemu CentOS w najnowszej wersji x86_64. Obraz płyty instalacyjnej można pobrać ze strony <http://www.centos.org>.
2. Do późniejszej aktualizacji systemu (pobrania aktualnych pakietów) zalecane jest przygotowanie dostępu do Internetu.
3. W BIOS komputera ustawiamy źródło uruchamiania (*ang. boot*) systemu z napędu DVD (szczegóły w dokumentacji technicznej komputera albo jego płyty głównej).
4. Wkładamy płytę do napędu i uruchamiamy komputer.
5. Po uruchomieniu płyty zobaczymy ekran powitalny instalatora CentOS.
6. Należy pamiętać, że dalej opisany proces instalacji może się różnić w szczegółach w zależności od konkretnie posiadanej wersji systemu.
7. Wybieramy *Install or upgrade an existing system* (opcję wybieramy strzałkami góra/dół i potwierdzamy klawiszem ENTER). Jądro startowe (*ang. startup kernel*) systemu zostanie załadowane i uruchomi się instalator.
8. Następnym etapem instalacji jest weryfikacja (sprawdzenie) nośników instalacyjnych. Jeśli jesteśmy pewni, że płyta z której instalujemy system jest prawidłowo nagrana możemy pominąć weryfikację wybierając SKIP. W przeciwnym wypadku wybieramy OK i czekamy na zakończenie testu nośnika.
9. Zobaczymy ekran powitalny, klikamy Next.

10. Wybieramy język systemu Polish (polski) i klikamy Next.
11. Wybieramy układ klawiatury polska i klikamy Dalej.
12. Wybieramy Utwórz indywidualny układ i klikamy Dalej.
13. Przechodzimy do programu partycjonującego dysk twardy:
 - a) Kasujemy wszystkie istniejące partycje (jeśli jakiegokolwiek były) i dzielimy nasz dysk twardy na partycje. Tworzymy dwa rodzaje partycji: podstawową dla /boot oraz rozszerzoną LVM dla pozostałych partycji.
 - b) Klikamy Nowa, następnie wybieramy punkt montowania /boot, typ systemu plików ext3 oraz rozmiar 100MB.
 - c) Klikamy Nowa, następnie wybieramy physical volume (LVM) oraz rozmiar Wypełnij do maksymalnej dozwolonej wielkości.
 - d) Na liście utworzonych przez nas partycji wybieramy partycję LVM i klikamy LVM w górnym pasku przycisków. Zmieniamy Nazwę grupy woluminów na main i poprzez dolne polecenie Woluminy logiczne tworzymy pozostałe partycje systemu z odpowiednimi punktami montowania. Na partycjach LVM korzystamy z systemu plików ext3.
 - e) Po stworzeniu wszystkich potrzebnych partycji klikamy Dalej.
14. Instalujemy program startowy (*ang. bootloader*) GRUB, pozostawiamy opcje, które podpowiada nam instalator i klikamy Dalej.
15. Konfigurujemy interfejsy sieciowe. Szczegóły połączenia sieciowego należy ustalić z administratorem sieci. Dla stacji roboczych zalecane jest korzystanie z adresów dynamicznych DHCP. Dla maszyn pełniących funkcję serwera należy ustawić statyczny adres IP. W górnym oknie z listą interfejsów wybieramy odpowiedni i klikamy Edytuj. Wyłączamy Enable IPv6 support, a przy opcji Enable IPv4 support wybieramy Manual configuration, wpisujemy adres IP serwera oraz maskę. Po kliknięciu OK uaktywnią nam się pola Brama, Pierwszy DNS, Drugi DNS które musimy poprawnie wypełnić zgodnie z ustawieniami sieci. Następnie klikamy Dalej.
16. Wybieramy strefę czasową Europa/Warszawa. Zaznaczamy Zegar systemowy używa UTC. Klikamy Dalej.
17. Ustawiamy hasło dla użytkownika root (głównego administratora). Klikamy Dalej.
18. Jesteśmy w miejscu gdzie konfigurujemy instalowane pakiety. Odznaczamy Biuro i praca biurowa. Wybieramy Dostosuj teraz. Klikamy Dalej.
19. Wyłączamy:
 - a) Środowisko graficzne / Środowisko graficzne GNOME
 - b) Aplikacje / Graficzny Internet
 - c) Serwery / Obsługa drukowania
 - d) Podstawowy system / Java
 - e) Podstawowy system / Obsługa sieci dial-up
 - f) Podstawowy system / System X WindowKlikamy Dalej.

20. Instalator sprawdzi zależności pakietów wybranych do instalacji, po sprawdzeniu klikamy Dalej w celu instalacji systemu.
21. Po zakończeniu instalacji automatycznie wysunie się dysk instalacyjny. Klikamy Uruchom ponownie.
22. Komputer uruchomi się ponownie.
23. Agent instalacji pozwoli nam wstępnie skonfigurować system. Zostajemy po informowani o możliwościach Agenta oraz Licencji na oprogramowanie instalowane razem z systemem. Na pierwszych dwóch ekranach klikamy Do przodu.
24. Opcję zapory sieciowej ustawiamy na Wyłączony.
25. W polu SELinux wybieramy Zezwalanie.
26. Ustawiamy poprawną datę i czas.
27. W oknie Hardware profile zostawiamy standardowe ustawienie i klikamy Do przodu, a następnie No, do not send.
28. Sugerujemy nie tworzyć dodatkowych użytkowników w tym momencie. Proces instalacji i konfiguracji będziemy mogli wykonać przy pomocy użytkownika root. Potwierdzamy klikając Do przodu a następnie Dalej.
29. Mamy możliwość sprawdzenia poprawnej instalacji i konfiguracji karty dźwiękowej. W przypadku instalacji serwera nie jest to konieczne.
30. Proces instalacji tzw. „czystego” systemu został zakończony. Należy teraz bezzwłocznie dokończyć konfigurację oraz zaktualizować pakiety składowe systemu do najnowszych wersji. W tym celu należy posłużyć się procedurą **Konfiguracja startowa systemu Linux CentOS**.

1.3. Konfiguracja bezpiecznego terminala SSH

1. Generowanie pary kluczy do autoryzacji

- a) W systemie Windows klucze do połączeń SSH generujemy za pomocą programu puttygen. Po włączeniu programu wybieramy przycisk Generate i wykonujemy dowolne ruchy myszką na szarym polu w oknie programu. Kiedy pasek postępu dojdzie do końca nasz klucz zostanie wygenerowany i będzie można zapisać go do pliku. Zaleca się ustawienie hasła do klucza podczas zapisywania go do pliku.

- b) W systemie Linux klucz generujemy za pomocą polecenia:

```
# ssh-keygen
```

Więcej na ten temat w dokumentacji:

```
# man ssh-keygen
```

2. Plik zaufanych kluczy authorized_keys

W tym pliku znajdującym się w katalogu użytkownika w podkatalogu .ssh znajdują się klucze prywatne użytkowników uprawnionych do logowania się na danego użytkownika bez podawania hasła. Do tego pliku wklejamy wcześniej wygenerowany klucz w formacie:

```
ssh-rsa klucz_użytkownika komentarz
```

Jako komentarz najlepiej wpisać e-mail osoby, aby można było w przyszłości łatwo

zidentyfikować klucze i szybko usunąć wpisy dla osób, które straciły uprawnienia.

3. Logowanie na serwer za pomocą klucza

Aby zalogować się na serwer przy użyciu wcześniej wygenerowanego klucza używamy programu puTTY, w którym w parametrach Connection → SSH → Auth w polu Private key file for authentication wskazujemy plik z wygenerowanym kluczem prywatnym.

4. Konfiguracja forward agenta

Za pomocą programu puTTY istnieje możliwość forwardowania klucza, czyli podawania go dalej przy logowaniu się na kolejne serwery z jednej konsoli. Warunkiem jest żeby na docelowym serwerze również znajdował się nasz klucz prywatny. Aby uruchomić tą funkcjonalność zaznaczamy pole Allow agent forwarding w parametrach programu putty: Connection → SSH → Auth.

1.4. Konfiguracja startowa systemu CentOS

1. Logujemy się na konsole systemową (tzw. terminal) jako użytkownik root (hasło użytkownika root jest ustalane podczas instalacji).
2. Sprawdzamy, czy komputer ma prawidłowy dostęp do Internetu. Możemy to wykonać np. przy pomocy komendy:

```
# ping www.onet.pl
PING www.onet.pl (213.180.130.200) 56(84) bytes of data.
64 bytes from flvirt.onet.pl (213.180.130.200): icmp_seq=1 ttl=59 time=13.4 ms
64 bytes from flvirt.onet.pl (213.180.130.200): icmp_seq=2 ttl=59 time=11.5 ms
```

Test można przeprowadzić również na inne adresy.

3. Aktualizujemy pakiety składowe systemu do najnowszych wersji:

```
# yum upgrade
```

Program yum wybierze pakiety, które wymagają aktualizacji, sprawdzi ich zależności, pobierze te pakiety i zainstaluje. Oczywiście przed aktualizacją systemu należy posiadać aktualną kopię bezpieczeństwa. W tym przypadku, aktualizujemy tzw. „czysty” system i kopię bezpieczeństwa pomijamy.

4. Dostęp do repozytoriów jest zwykle realizowany przez protokół HTTP (port TCP 80) albo RSYNC (port TCP 873). Jeżeli dostęp do repozytorium jest możliwy tylko przez serwer proxy to należy ustawić prawidłowo zmienne środowiskowe http_proxy i RSYNC_PROXY. Najłatwiej to zrobić przez stworzenie pliku /etc/profile.d/proxy.sh:

```
export http_proxy="http://192.168.1.5:8080"
export RSYNC_PROXY="192.168.1.5:8080"
```

Oczywiście powyższy adres IP i port jest przykładowy i musi być zgodny z lokalną konfiguracją sieci. Stworzenie powyższego pliku spowoduje, że system automatycznie ustawi powyższe zmienne środowiskowe po zalogowaniu się administratora.

5. Instalujemy bardzo przydatne narzędzie do zarządzania plikami Midnight Commander:

```
# yum install mc
```

6. Po skończeniu aktualizacji zalecany jest restart systemu. Restart wykonujemy przez naciśnięcie klawiszy CTRL-ALT-DEL na klawiaturze przyłączonej bezpośrednio do komputera. Zawsze przed użyciem kombinacji należy sprawdzić do jakiego komputera

jest podłączona klawiatura!

7. Usuwamy środowisko GCJ dla Java (jeżeli jest zainstalowane). Jest ono niekompatybilne z Java Sun.

```
# yum remove java-1.5.0-gcj*
```

Jeżeli niepoprawna Java była zainstalowana to system wyświetli ewentualne zależności pomiędzy pakietami. Potwierdzamy wtedy deinstalację wybierając y i naciskając ENTER.

8. Konfigurujemy dostęp do repozytorium rpm.finn.pl.

- a) Dostęp do rpm.finn.pl jest chroniony autoryzacją przy pomocy użytkownika i hasła. Dane te zostały zamieszczone na forum CELAB.

- b) Pobieramy i instalujemy pakiet finn-fedora z przygotowaną konfiguracją dostępową do repozytorium rpm.finn.pl:

```
# cd /root
# rsync -v rsync://celab@rsync.finn.pl/produkty-serwer/finn-rpm/fedora/RPMS/finn-fedora-* /root
# ls finn-fedora-*
# rpm -i finn-fedora-(wersja_pakietu)
```

Usługa rsync pracuje na porcie 873. Jeżeli dostęp do Internetu jest ograniczany tzw. ścianą ogniową (*ang. firewall*) to należy uzgodnić z administratorem sieci otwarcie w/w portu. Repozytorium dostępne jest również przez alternatywny protokół http na porcie 80. Do pobierania plików przez protokół http można użyć polecenia:

```
# wget http://celab:password@rpm.finn.pl/fedora/RPMS/finn-fedora-(wersja_pakietu)
# rpm -i finn-fedora-(wersja_pakietu)
```

- c) Ustawiamy użytkownika i hasło w konfiguracji repozytorium /etc/yum.repos.d/finn-fedora.repo. Do edycji pliku można wykorzystać mc (Midnight Commander) i klawisz funkcyjny F4. Korektę należy wprowadzić w linii:

```
baseurl=http://celab:password@rpm.finn.pl/fedora/RPMS/
```

- d) Sprawdzamy poprawność konfiguracji i aktualizujemy pakiety:

```
# yum upgrade
```

Jeżeli aktualizacja przebiegła poprawnie to znaczy, że dostęp do repozytorium działa prawidłowo.

9. Instalujemy środowisko Java:

- a) Instalujemy aktualne pakiety Java Sun przeznaczone dla naszej architektury sprzętowej.

```
# yum install jdk
```

Uwaga! Pliki oznaczone sygnaturą amd64 przeznaczone są również dla 64-bitowych procesorów Intel.

- b) Instalujemy pakiet narzędziowy javy.

```
# yum install jpackage-utils
```

- c) Po instalacji Javy Sun instalujemy pakiet niezbędny do ustawienia wszelkich zmiennych systemowych potrzebnych do prawidłowej pracy.

```
# yum install java-1.6.0-sun-compat
```

- d) Wykonujemy polecenie odbudowujące powiązania pomiędzy bibliotekami. Brak wykonania polecenia objawia się nieprzewidywalnym zachowaniem środowiska Java i jest trudne do zdiagnozowania. Dlatego polecenie należy koniecznie wykonać.

```
# /etc/cron.daily/prelink
```

10. Instalujemy pakiety z środowiskiem narzędziowym FINN:

```
# yum install finn-admin
```

11. Konfigurujemy przekierowanie poczty z konta root na adres e-mail administratora. Edytujemy w tym celu plik `/etc/aliases`. Po poprawieniu pliku należy wykonać polecenie:

```
# newaliases
```

12. Instalujemy pakiet `finn-common`

```
# yum install finn-common
```

Pakiet ten dostarcza skrypty niezbędne do pobrania i instalacji aplikacji CELAB, oraz instaluje menu ułatwiające wykonywanie podstawowych czynności administracyjnych na serwerze.

1.5. Instalacja i konfiguracja serwera bazy danych PostgreSQL

1. Nowo instalowany system będzie współpracował z bazą PostgreSQL 8.2. System CELAB **nie współpracuje** z nowszymi wersjami PostgreSQL (8.3, 8.4 i wyższe).
2. Instalujemy pakiety z oprogramowaniem zarządzającym relacyjną bazą danych PostgreSQL 8.2. Pakiety należy pobrać zgodnie z wersją i architekturą zainstalowanego systemu operacyjnego ze strony <http://yum.pgsqlrpms.org/8.2/>. Zalecamy instalację pakietu `pgdg-centos-8.2-4.noarch.rpm`, a następnie skorzystanie z programu `yum` w celu zainstalowania pakietów `postgresql`, `postgresql-devel`, `postgresql-libs`, `postgresql-server`.
3. Automatyczna konfiguracja serwera baz danych PostgreSQL

Za pomocą skryptu `finn-db-init` możemy skonfigurować i zainicjować serwer baz danych. Skrypt wykonuje półautomatycznie konfigurację zawartą w punktach od 3 do 7. Składnia polecenia jest następująca:

```
[root@serwer ~]# finn-db-init -h
Uzycie: /usr/sbin/finn-db-init [-p] [-D katalog] [-i] [-w]
-p - generowanie pliku /etc/profile.d/pgsql.sh
-D - lokalizacja bazy danych
-i - inicjalizuje wcześniej baze danych (initdb)
-w - zapisz zmiany do plików konfiguracyjnych (nie wykonuje kopii!!!)
Skrypt poprawia skrypty konfiguracyjne serwera bazy danych PostgreSQL
zgodnie z zaleceniami FINN.
Skrypt testowany na systemie Fedora Linux.
Przy nowej instalacji sugerowane parametry to: -p -i -w
```

4. Tworzymy pliki konfiguracyjne.

```
# echo "export PGUSER=postgres" > /etc/profile.d/pgsql.sh
```

5. Kasujemy zawartość katalogu gdzie ma być stworzona baza danych.

```
# rm -rf /var/lib/pgsql/data/*
```

Uwaga! Operacja jest nieodwracalna i należy mieć pewność, że kasujemy poprawny katalog.

6. Inicjujemy bazę danych.

```
# su postgres -c "initdb --pgdata=/var/lib/pgsql/data --encoding=UTF-8 --locale=pl_PL.UTF-8 --auth=md5 --username=postgres --pwprompt"
```

Uwaga! Należy zachować hasło administratora baz danych podane przy inicjacji bazy! Standardowe hasło to sql8.

7. Wprowadzamy zmiany w pliku konfiguracyjnym /var/lib/pgsql/data/postgresql.conf:

a) Nasłuchiwanie na wszystkich interfejsach i na domyślnym porcie 5432:

```
listen_addresses = '*'
port = 5432
```

b) Obsługa większych baz danych:

```
max_fsm_relations = 5000
```

c) Przekierowanie tzw. logów do katalogu /pg_log:

```
log_destination = 'stderr' # Logowanie na standardowe wyjście
redirect_stderr = on # Tylko dla PostgreSQL 8.2 i starszych
```

d) Ewentualne przywrócenie domyślnej konfiguracji tworzenia plików z logami serwera (najlepiej skasować lub zakomentować parametry log_directory, log_filename, log_truncate_on_rotation, log_rotation_age, log_rotation_size). Niektóre dystrybucje (np. Fedora Core) ustawiają własne, mniej optymalne wartości.

e) Dokładniejsze rejestrowanie zdarzeń oraz błędnych i nieoptymalnych zapytań:

```
log_min_error_statement = warning # Rejestrowanie błędnych poleceń
log_min_duration_statement = 1000 # Rejestrowanie nieoptymalnych poleceń
log_connections = on
log_disconnections = on
log_line_prefix = '%t %d [%p]: [%l] '
escape_string_warning = off
```

f) Automatyczne porządkowanie bazy danych:

```
stats_row_level = on # Tylko dla PostgreSQL 8.2 i starszych
autovacuum = on
```

8. Wprowadzamy zmiany w pliku konfiguracyjnym /var/lib/pgsql/data/pg_ident.conf:

a) Na końcu pliku dodajemy:

```
serwer    root    postgres
```

9. Wprowadzamy zmiany w pliku konfiguracyjnym /var/lib/pgsql/data/pg_hba.conf:

a) Jako pierwszą regułę dodajemy:

```
local    all    all    ident serwer
```

b) Jako ostatnią dodajemy regułę dla sieci lokalnych (linię należy dostosować do konfiguracji lokalnej sieci):


```
host    all    all    192.168.0.0/16    md5
```

10. O ile w danej instalacji będzie wykorzystywany język R do obliczeń statystycznych, należy zainstalować rozszerzenie bazy PostgreSQL o nazwie PL/R zgodnie z dokumentacją zawartą na stronie <http://www.joeconway.com/plr/>.

1.6. Poprawa wydajności serwera baz danych PostgreSQL

1. Pamięć współdzielona (parametr `shared_buffers`) wymaga skonfigurowania odpowiednio jądra systemu operacyjnego. Należy w pliku `/etc/sysctl.conf` dodać (poprawić) następujące linie:

```
kernel.shmmax = 134217728
kernel.shmall = 2097152
```

Po zmianie parametrów jądra wymagane jest ponowne uruchomienie systemu operacyjnego.

2. Sprawdzamy aktualną konfigurację (niektóre wartości domyślne są dynamiczne, dlatego sprawdzamy komendą `SHOW`), np.:

```
# psql
postgres=# SHOW maintenance_work_mem;
maintenance_work_mem
-----
16MB
(1 row)
```

3. Modyfikujemy plik `/var/lib/pgsql/data/postgresql.conf`

```
shared_buffers = 64MB # Dla 1GB dla PostgreSQL i kernel.shmmax = 134217728
shared_buffers = 128MB # Dla 2GB dla PostgreSQL i kernel.shmmax = 268435456
shared_buffers = 320MB # Dedykowany serwer +4GB dla PostgreSQL i kernel.shmmax =
536870912

maintenance_work_mem = 40MB # Przyspieszenie VACUUMDB dla 2GB
maintenance_work_mem = 72MB # Przyspieszenie VACUUMDB dla 3+GB
work_mem = 2MB # Dla 2GB dla PostgreSQL
work_mem = 4MB # Dedykowany serwer +4GB tylko dla PostgreSQL
checkpoint_segments = 8 # Ogranicza częste checkpoints
checkpoint_segments = 16 # Dla serwerów wczytujących dużo danych (np. replikacje)
```

4. Partycję z danymi bazy danych (zwykle `/var/lib/pgsql`) montujemy z opcjami `noatime` oraz `data=writeback`. Przykładowa linia w pliku `/etc/fstab`:

```
/dev/mapper/serwer-pgsql /var/lib/pgsql ext3 defaults,noatime,data=writeback 0 0
```

1.7. Instalacja i konfiguracja serwera aplikacyjnego JBoss

1. Weryfikujemy dostępność środowiska Java.

```
# java -version
java version "1.6.0_06" (...)
```

2. Z serwera FTP `produkty.finn.pl` (użytkownik i hasło dostępu zostały podane na forum CELAB) pobieramy np. poleceniem `wget` pakiet `finn-jboss-4.0.3.1.8-3888.noarch.rpm`. Instalujemy pobrany pakiet poleceniem:

```
# rpm -i finn-jboss-4.0.3.1.8-3888.noarch.rpm
```

3. Korygujemy konfigurację w pliku `/etc/finn/finn.conf`. W zależności od posiadanej pamięci operacyjnej ustawiamy parametry maszyny wirtualnej Java:

- a) dla 64-bit i 4 GB:

```
JAVA_OPTS="-Xms512m -Xmx3500m -XX:PermSize=512m"
```

- b) dla 64-bit i 6 GB:

```
JAVA_OPTS="-Xms512m -Xmx5500m -XX:PermSize=512m"
```

- c) dla 64-bit i 8 GB:

```
JAVA_OPTS="-Xms512m -Xmx7000m -XX:PermSize=512m"
```

1.8. Wczytanie bazy danych PostgreSQL z pliku archiwalnego backup

1. Tworzymy nową, pustą bazę danych.

```
# createdb nazwa_bazy
```

2. Wczytujemy dane z pliku archiwalnego w formacie `*.backup`.

```
# pg_restore -F c -d nazwa_bazy nazwa_pliku.backup
```

3. Do poprawnego działania bazy danych ustawiamy zmienną `search_path`.

```
# psql nazwa_bazy
postgres=# ALTER DATABASE nazwa_bazy SET search_path=grfinn;
postgres=# \q
```

4. Weryfikujemy ustawienia sekwencji (*ang. sequences*) dla pól numerowanych automatycznie (tzw. identyfikatorów).

```
# psql nazwa_bazy
postgres=# SELECT finn_seq_create();
postgres=# SELECT finn_seq_set();
postgres=# \q
```

Weryfikacja sekwencji jest szczególnie istotna jeśli ingerujemy w bazę danych i ręcznie importujemy dane do tabel.

1.9. Instalacja i konfiguracja aplikacji CELAB

Procedura instalacji aplikacji CELAB wykorzystuje skrypty narzędziowe z pakietu `finn-common` i przed instalacją aplikacji należy zweryfikować ich aktualność.

Niniejsza procedura opisuje konfigurację zakładającą obsługę pojedynczej aplikacji CELAB. Aby uruchomić jednocześnie kilka aplikacji należy odpowiednio zmodyfikować pliki konfiguracyjne i wymaga to dodatkowej wiedzy na temat serwera aplikacji JBoss.

1. Sprawdzamy dostępność aktualizacji dla pakietu `finn-common`.

```
# yum upgrade finn-common
```

2. Edytujemy plik `/etc/finn/finn.conf` i ustawiamy:

- a) położenie repozytorium produktu oraz hasło dostępowe,

```
FINN_WEBSRC=rsync://celab@rsync.finn.pl/produkty-celab
FINN_WEBPASS=hasło
```

- b) nazwę bazy z danymi (lub kilka nazw oddzielonych spacjami),

```
FINN_PGDATABASES=piwet
```

- c) nazwę bazy z repozytorium plików (m.in. dla protokołu `webdav`).

FINN_PGDATABASESWEBDAV=piwet_pliki

3. Konfigurujemy aplikację:

a) Każda aplikacja ZHW pracuje na unikalnym adresie. Format adresacji to: <https://serwer.nazwazhw.finn.pl:8443>

b) Wirtualny host dla aplikacji ustawiamy w pliku /etc/finn/app-ff8/WEB-INF/jetty-web.xml . Plik powinien wyglądać następująco:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Configure PUBLIC "-//Mort Bay Consulting//DTD Configure//EN"
"http://jetty.mortbay.org/configure.dtd">
<Configure class="org.jboss.jetty.JBossWebAppContext">
  <Set name="VirtualHosts">
    <Array type="java.lang.String">
      <Item>192.168.1.155</Item>
      <Item>serwer.zhwkatowice.finn.pl</Item>
    </Array>
  </Set>
  <Call name="setInitParams">
    <Arg>
      <Map>
        <Entry>
          <Item>scheduler</Item>
          <Item>0 10 0/6 * * ?###importSlovníkiCbd###0
20 0 * * ?
###exportProbki###0 * * * * ?###cronMrap1</Item>
        </Entry>
      </Map>
    </Arg>
  </Call>
</Configure>
```

c) Linie

```
<Item>192.168.1.155</Item>
```

```
<Item>serwer.zhwkatowice.finn.pl</Item>
```

Zamieniamy na odpowiednie dla danej lokalizacji.

4. Wyświetlamy listę dostępnych aplikacji w repozytorium.

```
# finn-web-update -l
```

5. Pobieramy najnowszą wersję aplikacji i bazy wzorcowej z repozytorium.

```
# finn-web-update -w wersja_programu
```

```
# finn-web-update -w 8.1.03.c.13487
```

Uwaga! Powyższy format oznaczania wersji jest przykładowy i może ulec zmianie.

6. Wczytujemy do serwera PostgreSQL pobraną wyżej bazę wzorcową (z pliku w formacie archiwum PostgreSQL .backup).

```
# finn-db-template
```

7. Wczytujemy bazę z danymi np. przy pomocy procedury **Wczytanie bazy danych PostgreSQL z pliku archiwalnego backup**. Oczywiście punkt można pominąć jeżeli baza z danymi wcześniej już istniała na serwerze SQL.

8. Aktualizujemy bazy z danymi do nowej struktury na podstawie bazy wzorcowej.

```
# finn-db-migrate -d nazwa_bazy -t ff8u
```

```
np. # finn-db-migrate -d piwet -t ff8u
```

Uwaga! Powyższy format oznaczania bazy wzorcowej może ulec zmianie.

9. Instalujemy aplikację w środowisku aplikacyjnym JBoss.

```
# finn-web-install -f
```

Skrypt zapyta nas o informacje niezbędne do skonfigurowania aplikacji (adres serwera SQL, nazwy baz danych, nazwę i hasło użytkownika do bazy danych). Po podaniu tych informacji skrypt wygeneruje pliki potrzebne do uruchomienia aplikacji w standardzie J2EE.

10. Uruchamiamy serwer aplikacyjny JBoss.

```
# service finn-jboss start
```

11. Sprawdzamy poprawność uruchomienia serwera.

```
# service finn-jboss log-f
```

Każdy z powyższych skryptów ma możliwość wyświetlenia skróconego opisu użycia, realizowaną za pomocą przełącznika `-h`, na przykład:

```
# finn-db-template -h
```