FINN Sp. z o.o.
NIP 8321912565
REGON 731586439
KRS 0000023991 Kapitał zakł. 1 400 000 PLN
Sąd Rej. Łódź-Śródmieście XX Wydział KRS

Adres kontaktowy – Oddział w Łodzi
93-490 Łódź, ul. Pabianicka 159/161
tel. (42) 206 66 00, (42) 684 98 91
fax (42) 684 98 92
sekretariat@finn.pl,
http://www.finn.com.pl

F8WEB PROC

Opis techniczny procedur na okoliczność awarii oraz bieżących prac serwisowych

Autor: praca zespołowa pod redakcją Przemysława Sztocha

Historia zmian dokumentu:

Data	*Wersja*	*Opis*
2008-04-15	v1.0	Utworzenie struktury dokumentu.
2008-04-29	v1.2	Procedura inicjowania slotu HSM dla ESP.
2008-06-16	v1.3	Aktualizacja procedur dla osób korzystających z serwera proxy. Raportowanie z HSM. Wymiana klucza i certyfikatu SSL w JBoss oraz przekierowanie portów. Konfiguracja harmonogramu zadań automatycznych w aplikacji FINN. Większa przejrzystość w konfiguracji sprzętu sieciowego Cisco. Procedury skanowania.
2008-08-12	v1.4	Rozszerzono rozdział „Archiwizacja na płytę DVD, DVDRW”
2008-08-28	v1.5	Zmienione działanie repozytorium kursów (nowy rozdział 1.14 Konfiguracja automatycznej aktualizacji kursów e-learning).
2008-10-10	v1.6	Aktualizacja dokumentu do nowszej wersji systemu operacyjnego Fedora 9.
2008-12-22	v1.7	Aktualizacja procedury aktualizacji. Wydzielenie osobnego dokumentu F8WEB SKAN dla rozdziału Instalacja i konfiguracja skanera.
2009-01-29	v1.8	Aktualizacja dokumentu do pakietu finn-admin-1.3 oraz finn-common-1.5.
2009-02-02	v2.0	Aktualizacja dokumentu do pakietu finn-jboss4.
2009-03-02	v2.1	Poprawki wydajnościowe dla serwera baz danych PostgreSQL 8.2.
2009-06-08	v2.2	Poprawki wydajnościowe oraz zgodność z serwerem baz danych PostgreSQL 8.3.
2009-06-20	v2.3	Aktualizacja procedury instalacji. Dodanie źródła instalacji pakietów postgresql-*.
2010-01-05	v2.4	Aktualizacja procedur awaryjnych. Dodanie źródeł instalacji dla pakietów postgresql-8.4, aktualizacja instrukcji instalacji dla Fedora 12 i CentOS 5.4, dodatkowe repozytorium dla CentOS 5.4.
2010-03-04	v2.5	Korekta literówek. Nowe polecenie finn-db-info. Zalecenia dotyczące wydajności.
2010-09-10	v2.6	Aktualizacja pliku dla CentOS 5.5.
2011-03-29	v2.7	Aktualizacja instrukcji instalacji (nowe pakiety repozytorium rpm-finn). Aktualizacja opisu dostępnych parametrów skryptów aktualizacyjnych.
2011-09-15	v2.8	Aktualizacja instrukcji konfiguracji skrzynki pocztowej. Nowe polecenia instalacyjne dla pfe.
2012-01-03	v3.0	Aktualizacja procedur do CentOS 6.2, PostgreSQL 9.0.5
2012-01-30	v3.1	Szczegółowa instrukcja konfiguracji Komputera Komunikacyjnego
2012-06-15	v3.2	Poprawki firewall
2012-09-10	v3.3	Aktualizacja instrukcji instalacji i aktualizacji aplikacji FINN
2012-10-17	v3.4	Aktualizacja procedur instalacji Komputera Komunikacyjnego, PostgreSQL 9.1, poprawki procedur
2012-11-26	v3.5	Poprawki instalacji HSM sieciowego. Ogólne poprawki.
2013-03-08	v3.6	Aktualizacja procedur do CentOS 6.4, PostgreSQL 9.2
2014-10-17	v3.7	Aktualizacja procedur konserwacyjnych – serwer czasu ntpd
2015-02-03	v3.8	Aktualizacja sterowników HSM
2015-11-25	v3.9	Aktualizacja procedur do CentOS 6.7, wydzielenie dokumentu F8WEBPROC KK
2016-10-18	**v4.0**	**Aktualizacja procedur do CentOS 7 i PostgreSQL 9.4**

Spis treści

1. Instalacja, konfiguracja oraz aktualizacja

1.1. Instalacja Serwera z oprogramowaniem FINN

W celu przygotowania serwera należy wykonać procedury:

1. **Instalacja oprogramowania systemowego RHEL / Centos 7**
   a) Płyta instalacyjna w wersji 64-bitowej (serwery powinny być wyposażone w procesor o architekturze 64-bitowej.)
   b) Konfiguracja interfejsów sieciowych wg projektu sieci.
2. **Konfiguracja startowa systemu RHEL / Centos 7**
3. **Instalacja i konfiguracja serwera bazy danych PostgreSQL**
4. **Instalacja i konfiguracja serwera aplikacyjnego JBoss**
5. **Instalacja i konfiguracja aplikacji FINN**

1.2. Instalacja oprogramowania systemowego RHEL / Centos 7

1. Oprogramowanie powinno być instalowane na architekturze 64 bitowej. W związku z powyższym należy przygotować płytę instalacyjną **RHEL / Centos 7 x86_64**. Tylko architektura 64 bitowa jest w stanie prawidłowo i wydajnie obsłużyć więcej niż 3 GB pamięci. Do instalacji używamy wersji minimal-iso.

   Do pobrania np: http://centos.slaskdatacenter.com/7/isos/x86_64/CentOS-7-x86_64-Minimal-1511.iso
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 **RHEL / Centos 7**
6. Należy pamiętać, że dalej opisany proces instalacji może się różnić w szczegółach w zależności od konkretnie posiadanej wersji systemu.
7. Wybieramy **Install CentOS 7** (opcję wybieramy strzałkami góra/dół i potwierdzamy klawiszem **ENTER**). Jądro startowe (*ang. startup kernel)* systemu zostanie załadowane i uruchomi się instalator.
8. W lewej kolumnie wybieramy język systemu **Polski,** w prawej kolumnie język klawiatury **Polski (Polska)** klikamy **Kontynuuj**.
9. W grupie **SYSTEM** wybieramy **CEL INSTALACJI**
10. W opcji **Partycjonowanie** wybieramy **Konfiguracja partycjonowania użytkownika**
11. W lewej kolumnie wybieramy **+ (Dodaj nowy punkt montowania)**. Wybieramy punkt montowania /boot, wpisujemy żądaną pojemność 1024 i klikamy **Dodaj punkt montowania.**
12. Analogicznie tworzymy pozostałe partycje. Najważniejsze to / (katalog główny systemu operacyjnego) – 15GB, /home – 20GB, /var – 30GB, /var/lib/pgsql, /opt – ich wielkość dobieramy zależnie od pojemności dysków twardych.
13. W lewym górnym rogu klikamy przycisk **Gotowe** i potwierdzamy za pomocą przycisku **Zaakceptuj zmiany.**
14. W grupie **SYSTEM** wybieramy **SIEĆ I NAZWA KOMPUTERA**
15. Ustalamy nazwę hosta dla naszego systemu w oknie **Nazwa komputera**, np: sod-demourzad.finn.pl
16. W lewej kolumnie wybieramy odpowiednią kartę sieciową i w prawym dolnym rogu klikamy **Skonfiguruj...**
17. W zakładce **Ogólne** zaznaczamy **Automatyczne łączenie z tą siecią,kiedy jest dostępna.**
18. W zakładce **Ustawienia IPv4** z listy **Metoda** wybieramy Ręczne i za pomocą przycisku **Dodaj** konfigurujemy adres IP naszej karty sieciowej. W naszym przykładzie karta sieciowa ma adres 192.168.1.2/255.255.255.0, adres bramy sieciowej to 192.168.1.1, adres serwera DNS to 192.168.1.1, domena wyszukiwania finn.pl. Klikamy **Zapisz**. Karta sieciowa zostanie skonfigurowana i automatycznie włączona. Poniżej przykład jak powinna wyglądać poprawnie skonfigurowana i podłączona karta sieciowa. Jeśli wszystko się zgadza potwierdzamy operację za pomocą przycisku **Gotowe** w lewym górnym rogu ekranu.
19. Rozpoczynamy instalację systemu za pomocą przycisku **Rozpocznij instalację.**
20. Ustalamy hasło użytkownika root klikając przycisk **HASŁO ROOTA**. Potwierdzamy za pomocą przycisku **Gotowe**.
21. System rozpocznie instalację. Po zakończeniu automatycznie wysunie się dysk instalacyjny. Klikamy

Uruchom ponownie.

22. 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 7**

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 publiczne 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 prywantym.
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 usług systemowych – ntsysv

Serwery są zabezpieczone na wypadek zaniku zasilania. System skonfigurowany jest tak, że po ustalonym okresie czasu następuję automatyczne jego wyłączenie. Po powrocie zasilania system uruchamia się automatycznie. Za pomocą polecenia ntsysv można przejrzeć oraz zweryfikować jakie usługi zostaną automatycznie włączone podczas uruchomienia serwera a jakie pominięte.

1.5. Konfiguracja startowa systemu RHEL / Centos 7

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 f1virt.onet.pl (213.180.130.200): icmp_seq=1 ttl=59
   time=13.4 ms
   64 bytes from f1virt.onet.pl (213.180.130.200): icmp_seq=2 ttl=59
   time=11.5 ms
   ```
   Test można przeprowadzić również na inne adresy.
3. W związku z tym, że korzystaliśmy z płyty instalacyjnej wersji minimal musimy zainstalować niezbędne pakiety systemowe:
   ```
   # yum install -y cronie-noanacron mc man wget rsync postfix ntsysv ntp
   unzip zip file screen mailx telnet minicom dump yum-utils nfs-utils gcc
   make psmisc firewalld
   ```

4. Odinstalowujemy Anacron ponieważ jest zbędny.
   ```
   # yum remove -y cronie-anacron
   ```
5. Zmieniamy politykę selinuxa (wyłączamy go) edytując plik /etc/selinux/config
   ```
   # mcedit /etc/selinux/config
   ```
   Zmieniamy opcję SELINUX=enforcing na disabled

   Poprawnie plik powinien wyglądać następująco:

   ```
   # This file controls the state of SELinux on the system.
   # SELINUX= can take one of these three values:
   #     enforcing - SELinux security policy is enforced.
   #     permissive - SELinux prints warnings instead of enforcing.
   #     disabled - No SELinux policy is loaded.
   SELINUX=disabled
   # SELINUXTYPE= can take one of these two values:
   #     targeted - Targeted processes are protected,
   #     mls - Multi Level Security protection.
   SELINUXTYPE=targeted
   ```

   Restartujemy serwer.
6. 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.
7. 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.
8. 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!
9. Konfigurujemy dostęp do repozytorium **rpm.finn.pl**.
   a) Dostęp do **rpm.finn.pl** jest chroniony autoryzacją przy pomocy użytkownika i hasła. Należy uzyskać stosowne parametry dostępu.
   b) Pobieramy i instalujemy pakiet **finn-rpm** z przygotowaną konfiguracją dostępową do repozytorium **rpm.finn.pl**:
   c) Sprawdzamy najnowszą wersję pakietu finn-rpm w repozytoriu, rpm.finn.pl, następnie pobieramy ją za pomocą wget:
      ```
      # cd /root
      # wget http://finn8web:bungi@rpm.finn.pl/rhel7/x86_64/finn-rpm-rhel7-1.4-27010.noarch.rpm
      # rpm -ivh finn-rpm-rhel7-1.4-27010.noarch.rpm
      ```
   d) Sprawdzamy poprawność konfiguracji i aktualizujemy pakiety:
      ```
      # yum upgrade
      ```
      Jeżeli aktualizacja przebiegła poprawnie to znaczy, że dostęp do repozytorium działa prawidłowo.
10. 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.

11. Instalujemy pakiety z środowiskiem narzędziowym FINN:
```
# yum install finn-admin
```
12. 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
```
13. Instalujemy pakiet finn-common
```
# yum install finn-common
```
Pakiet ten dostarcza skrypty niezbędne do pobrania i instalacji aplikacji FINN, oraz instaluje menu ułatwiające wykonywanie podstawowych czynności administracyjnych na serwerze.
14. Konfiguracja firewalld

Przykładowa konfiguracja firewalld w pliku /etc/firewalld/zones/public.xml
```
<?xml version="1.0" encoding="utf-8"?>
<zone>
  <short>Public</short>
  <description>For use in public areas. You do not trust the other computers on networks to
not harm your computer. Only selected incoming connections are accepted.</description>
  <service name="dhcpv6-client"/>
  <service name="ssh"/>
  <port protocol="tcp" port="80"/>  <!-- http /-->
  <port protocol="tcp" port="111"/> <!-- nfs /-->
  <port protocol="tcp" port="443"/> <!-- https /-->
  <port protocol="tcp" port="873"/> <!-- rsync /-->
  <port protocol="tcp" port="2049"/> <!-- nfs /-->
  <port protocol="tcp" port="5432"/> <!-- postgresql /-->
  <port protocol="tcp" port="8080"/> <!-- proxy (squid) /-->
  <port protocol="udp" port="69"/> <!-- tftp /-->
  <port protocol="udp" port="111"/> <!-- nfs /-->
  <port protocol="udp" port="161"/> <!-- snmp /-->
  <port protocol="udp" port="2049"/> <!-- nfs /-->
  <masquerade/>
  <forward-port to-port="8000" protocol="tcp" port="80"/>
  <forward-port to-port="8443" protocol="tcp" port="443"/>
</zone>
```

Po skonfigurowaniu firewalld uruchamiamy proces i dodajemy do autostartu
```
# systemctl enable firewalld
# systemctl start firewalld
```
15. W zależności od przeznaczenia należy zainstalować i skonfigurować oprogramowanie aplikacyjne oraz skonfigurować system archiwizacji. W tym celu należy posłużyć się procedurami:
    a) **Instalacja i konfiguracja serwera bazy danych PostgreSQL**
    b) **Instalacja i konfiguracja serwera aplikacyjnego JBoss**
    c) **Instalacja i konfiguracja aplikacji FINN 8 SQL**

1.6. Dodatkowe repozytoria pakietów

Przy kolejnych krokach instalacji systemu możemy potrzebować pakietów, które nie są dostępne w oficjalnych repozytoriach plików CentOS/Red Hat. Dlatego też zaleca się konfigurację dodatkowych repozytoriów EPEL.

1. Pobieramy najnowszą wersję pakietu z repozytorium EPEL
```
# wget https://dl.fedoraproject.org/pub/epel/epel-release-latest-
7.noarch.rpm
```
2. Instalujemy repozytorium
```
# rpm -ivh epel-release-latest-7.noarch.rpm
```

3. Aktualizujemy repozytoria
   ```
   # yum check-update
   ```

1.7. Instalacja i konfiguracja serwera bazy danych PostgreSQL

Uwaga! Opisana procedura dotyczy PostgreSQL 9.4

1. Instalujemy pakiety z oprogramowaniem zarządzającym relacyjną bazą danych. Do instalacji pakietów nie potrzebujemy dodatkowych repozytoriów. Pakiety PostgreSQL 9.4 umieszczone są w repozytorium finn-rpm.
2. ```
   # yum install postgresql94 postgresql94-server postgresql94-tsdict
   postgresql94-contrib pwgen
   ```
3. Automatyczna konfiguracja serwera baz danych PostgreSQL

   Za pomocą skryptu **postgresql-9.4-initdb** możemy skonfigurować i zainicjować serwer baz danych. Składnia polecenia jest następująca:

   [root@sod-demourzad ~]# postgresql-9.4-initdb -h
   ```
   Uzycie: /usr/sbin/postgresql-9.4-initdb [-D katalog] [-m mem] [-b] [-r] [-w] [-c]
   -D - lokalizacja bazy danych (np. /var/lib/pgsql/9.4/data)
   -m - pamięć serwera w MB dedykowana dla PostgreSQL (domyślnie 993)
   -b - skonfiguruj serwer dla pgBarman (archiwizacja)
   -r - skonfiguruj serwer dla pgBadger (raportowanie)
   -w - zapisz zmiany do plikow konfiguracyjnych
   -c - tryb kompatybilności (wymagany przez FINN 8 SQL SOD)
   Skrypt poprawia skrypty konfiguracyjne serwera bazy danych PostgreSQL
   zgodnie z zaleceniami FINN.
   ```

4. Inicjujemy bazę danych i wprowadzamy niezbędne zmiany w plikach konfiguracyjnych. Przykład:
   ```
   # postgresql-9.4-initdb -D /var/lib/pgsql/9.4/data -m 2048 -w -c
   ```

   Hasło dla użytkownika postgres jest generowane i zapisywane w pliku /root/.pgpass

   Wygenerowane hasło wpisujemy w pliku /etc/finn.conf w parametrze FINN_PGPASS zastępując hasło domyślne.
5. Uruchomienie i automatyczny start serwera PostgreSQL przy starcie systemu
   ```
   # systemctl enable postgresql-9.4
   # systemctl start postgresql-9.4
   ```

6.

1.8. Poprawa wydajności serwera baz danych PostgreSQL

1. Pamięć współdzielona (parametr **shared_buffers**) wymaga skonfigurowania odpowiednio jądra systemu operacyjnego. Odpowiadają za to wpisy w pliku **/etc/sysctl.conf:**
   ```
   kernel.shmmax = 134217728
   kernel.shmall = 2097152
   ```
   Po zmianie parametrów jądra wymagane jest ponowne uruchomienie systemu operacyjnego. Systemy CentOS i RHEL serii 7 mają standardowo ustawione te parametry w prawidłowy sposób, więc nie ma potrzeby ich zmiany.
2. Jeżeli to konieczne 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)
   ```
   Musimy jednak pamiętać, że komenda psql wymaga do działania uruchomionego wcześniej serwera PostgreSQL.
   ```

3. Modyfikujemy plik **/var/lib/pgsql/data/9.4/postgresql.conf**

```
shared_buffers = 64MB # Dla 1GB dla PotsgreSQL i kernel.shmmax = 134217728
shared_buffers = 128MB # Dla 2GB dla PotsgreSQL i kernel.shmmax = 268435456
shared_buffers = 320MB # Dedykowany serwer +4GB dla PotsgreSQL 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 PotsgreSQL
work_mem = 4MB # Dedykowany serwer +4GB tylko dla PotsgreSQL
checkpoint_segments = 8 # Ogranicza częste chceckpointy
checkpoint_segments = 16 # Dla serwerów wczytujących dużo danych (np. replikacje)
```

   Jeżeli inicjowaliśmy bazę danych poleceniem **postgresql-9.4-initdb** to powyższe parametry powinny być automatycznie skorygowane. Nie mniej wskazana jest ich weryfikacja w stosunku do konkretnie posiadanej platformy sprzętowej (RAM, dyski, tryb pracy RAID itp.).

4. Partycję z danymi bazy danych (zwykle **/var/lib/pgsql**) montujemy z opcją **noatime**. Przykładowa linia w pliku **/etc/fstab**:

```
/dev/mapper/centos-var_lib_pgsql /var/lib/pgsql          xfs     defaults,noatime        0 0
```

   Opcja **noatime** pozwala na nie zapisywanie czasu dostępu do plików co prowadzi do zmniejszenia ilości operacji wykonywanych na dysku twardym, przez co również do wzrostu wydajności.

5. Jeżeli baza danych jest posadawiana na macierzy RAID najbardziej zalecany jest tryb RAID10. Ze względów wydajnościowych przy dyskach SATA nie zalecamy stosowania RAID5. Przy sprzętowych kontrolerach RAID niezbędne jest włączenie tzw. write cache.

1.9. Instalacja i konfiguracja serwera aplikacyjnego JBoss4

1. Weryfikujemy dostępność środowiska Java.

```
# java -version
java version "1.6.0_24" (...)
```

2. Sprawdzamy dostępność aktualizacji pakietu z serwerem aplikacyjnym JBoss.

```
# yum upgrade finn-jboss4*
```

   Jeśli pakietu n:qie wykryto, instalujemy go.

```
# yum install finn-jboss4*
```

3. Korygujemy konfigurację w pliku **/etc/finn/finn.conf**. W zależności od posiadanej pamięci operacyjnej ustawiamy parametry maszyny wirtualnej Java:
   a) 32-bit i 2GB (lub więcej):
      `JAVA_OPTS="-Xms512m -Xmx1700m -XX:PermSize=128m"`
   b) dla 64-bit i 2 GB:
      `JAVA_OPTS="-Xms512m -Xmx1700m -XX:PermSize=256m"`
   c) dla 64-bit i 4 GB:
      `JAVA_OPTS="-Xms512m -Xmx3500m -XX:PermSize=512m"`
   d) dla 64-bit i 6 GB:
      `JAVA_OPTS="-Xms512m -Xmx5500m -XX:PermSize=512m"`
   e) dla 64-bit i 8 GB:
      `JAVA_OPTS="-Xms512m -Xmx7000m -XX:PermSize=512m"`

1.10. Instalacja i konfiguracja aplikacji FINN

Procedura instalacji aplikacji FINN wykorzystuje skrypty narzędziowe z pakietu **finn-common** i przed instalacją aplikacji należy zweryfikować ich aktualność. Aplikacja wymaga również odpowiedniej wersji serwera aplikacji JBoss z pakietu **finn-jboss4**. Wyżej wymienione pakiety znajdują się w repozytoriach i można je aktualizować przy pomocy procedury **Aktualizacja pakietów oprogramowania wg repozytoriów**. Jeżeli pakiety nie zostały zainstalowane to należy je zainstalować zgodnie z odpowiednią procedurą.

Aplikacja wymaga również działającego serwera baz danych PostgreSQL (pakiety **postgres***). Serwer SQL może być zainstalowany na:

- tym samym komputerze - dostępny lokalnie pod adresem 127.0.0.1,
- innym (np. dedykowanym do obsługi baz danych) komputerze - dostępny zdalnie pod określonym adresem IP

poprzez sieć TCP/IP.

Niniejsza procedura opisuje konfigurację zakładającą obsługę pojedynczej aplikacji FINN. Aby uruchomić jednocześnie kilka aplikacji należy odpowiednio zmodyfikować pliki konfiguracyjne i wymaga 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://finn8web@rsync.finn.pl/produkty-finn8web
   FINN_WEBPASS=hasło
   ```
   b) położenie repozytorium kursów e-learning oraz hasło dostępowe,
   ```
   FINN_KURSYSRC=rsync://lic00000@rsync.finn.pl/produkty-kursy-lic00000
   FINN_KURSYPASS=hasło
   ```
   c) adres serwera SQL (tylko wtedy jeżeli **nie jest lokalny**),
   ```
   FINN_PGHOST=192.168.1.20
   ```
   d) hasło dla użytkownika postgres
   ```
   FINN_PGPASS=sql8
   ```
   e) nazwę bazy z danymi (lub kilka nazw oddzielonych spacjami),
   ```
   FINN_PGDATABASES=demogmina
   ```
   f) nazwę bazy z repozytorium plików (m.in. dla protokołu webdav).
   ```
   FINN_PGDATABASESWEBDAV=demogmina_pliki
   ```
3. Kopiujemy przykładową konfigurację aplikacji (przed kopiowaniem należy mieć pewność, że docelowy folder nie istnieje).
   ```
   # rm -rf /etc/finn/app-ff8
   # cp -a /etc/finn/app-ff8-example /etc/finn/app-ff8
   ```
4. Edytujemy plik **/etc/finn/app-ff8/WEB-INF/context.xml** i ustawiamy zadania do automatycznego i cyklicznego wykonywania przez serwer aplikacyjny:
   ```
   <Context>
   <Parameter name="scheduler" value="0 * * * * ?###cronMrap1###0 * * * * ?
   ###rejestrujPoczte###0 * * * * ?###cronEcp###0 30 0 * * ?###cronDaily" />
   </Context>
   ```
5. Wyświetlamy listę dostępnych aplikacji w repozytorium.
   ```
   # finn-web-update -l
   ```
6. Pobieramy najnowszą wersję aplikacji i bazy wzorcowej z repozytorium.
   ```
   # finn-web-update -w wersja_programu
   # finn-web-update -w 8.1.11.s.28043 -p
   ```
   Uwaga! Powyższy format oznaczania wersji jest przykładowy i może ulec zmianie.
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.
   Procedura aktualizacji opisana jest w punkcie **1.18 Aktualizacja aplikacji FINN**
9. W pliku **/home/services/finn/jboss/server/finn/deploy/jboss-web.deployer/server.xml** konfigurujemy porty działania aplikacji, **redirectPort** poprawiamy na 443 np.:
   ```
   <Connector port="8000" maxThreads="250" maxHttpHeaderSize="8192" maxPostSize="-1"
     emptySessionPath="true" protocol="HTTP/1.1" useBodyEncodingForURI="true"
     enableLookups="false" redirectPort="8443" acceptCount="100"
     connectionTimeout="20000" disableUploadTimeout="true" />
   ```
10. 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.

11. W pliku /etc/hosts do linii 127.0.0.1 dopisujemy hostname serwera, np:
    ```
    127.0.0.1   localhost localhost.localdomain apl2
    ```
12. Uruchamiamy serwer aplikacyjny JBoss.
    ```
    # service finn-jboss start
    ```
13. 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-migrate -h
```

1.11. Instalacja platformy PFE.

Procedura instalacja platformy pfe wykorzystuje skrypty z pakietu finn-common. Przed instalacją musimy upewnić się, że mamy wersję pakietu finn-common conajmniej 1.6-21724. Pakiet możemy zaktualizować za pomocą polecenia:

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

1. Pobieramy i instalujemy aplikację pfe na serwer SOD.
   ```
   # finn-pfe-upgrade
   ```
2. Po wykonaniu polecenia instalacyjnego program kopiowany jest w odpowiednie miejsce i automatycznie uruchamiany. Log z uruchamiania się programu znajduje się w standardowym logu finn-jboss - /var/log/finn/jboss-finn/server.log
7.

1.12. Wymiana klucza i certyfikatu SSL w JBoss oraz przekierowanie portów

1. Należy przygotować klucz prywatny i certyfikat w formacie P12.

   Jeżeli mamy klucz i certyfikat tylko w postaci plików PEM to należy je skonwertować na format P12. Można do tego celu wykorzystać komendę **openssl**.

   a) Warto je zweryfikować komendami:
      ```
      # openssl x509 -in serwer.crt -text
      # openssl rsa -in serwer.key -text
      ```
   b) Następnie generujemy P12. Zapamiętujemy podane hasło eksportowe (będzie potrzebne w następnym punkcie).
      ```
      # openssl pkcs12 -inkey serwer.key -in serwer.crt -export -out
      serwer.p12
      ```
2. Tworzymy magazyn kluczy w standardzie Java za pomocą keytool.

   Magazyn kluczy powinien nazywać się np. tak jak nazwa dns domeny. Dodatkowo dodajemy kropkę na początku, aby plik był traktowany jako ukryty. Jako hasło wejściowe podajemy hasło do P12 (patrz poprzedni punkt). Jako hasło wyjściowe podajemy takie samo jak do P12. Należy je zapamiętać ponieważ będzie użyte później przy konfiguracji aplikacji (punkt 1.11.3).

   ```
   # keytool -importkeystore -srckeystore serwer.p12 -srcstoretype pkcs12
   -srcstorepass naszeHaslo -destkeystore .serwer.keystore -deststorepass
   naszeHaslo
   ```
3. Należy sprawdzić czy w konfiguracji prawidłowo jest określony pod jakim portem zewnętrznym SSL widziany jest serwer aplikacji (parametr redirectPort):

   ```
   <Connector port="8000" maxThreads="250" maxHttpHeaderSize="8192" maxPostSize="-1"
         emptySessionPath="true" protocol="HTTP/1.1" useBodyEncodingForURI="true"
         enableLookups="false" redirectPort="8443" acceptCount="100"
         connectionTimeout="20000" disableUploadTimeout="true" />
   ```

   Oczywiście powyższa linia musi ściśle współgrać z przekierowaniem o którym mowa w następnym punkcie.
4. Użytkownicy łącząc się z aplikacją powinni robić to przez standardowe porty sieciowe (TCP 80 dla usługi http i TCP 443 dla usługi https).

   W systemie Linux serwer aplikacji JBoss nie ma uprawnień do nasłuchiwania na niskich portach (poniżej

1024). Z tego powodu JBoss nasłuchuje na portach wysokich (standardowo jest to TCP 8000 dla usługi http i TCP 8443 dla usługi https). Korzystanie z wysokich portów jest **niezalecane**. Dlatego należy skonfigurować odpowiednie przekierowania (z portu 80 na 8000 i z portu 443 na 8443).

Do tego celu można wykorzystać **firewalld** (firewall wbudowany w Linux). Przykładowa konfiguracja przekierowania została opisana w procedurze **Instalacja Serwera z oprogramowaniem FINN**.

1.13. Konfiguracja harmonogramu zadań automatycznych w aplikacji FINN

Aplikacja FINN pracująca pod kontrolą serwera aplikacji JBoss jest wyposażona we wbudowany harmonogram zadań (tzw. cron). Konfiguracja harmonogramu jest standardowo przechowywana w pliku **/etc/finn/app-*ff8*/WEB-INF/context.xml**.

Do określenia listy zadań, które mają zostać automatycznie wykonywane w podanych interwałach czasowych służy parametr **scheduler**. Należy pamiętać, że parametr **scheduler** można podać tylko raz, a wszystkie zaplanowane zadania muszą zostać podane w jednej linii.

Dla funkcji rejestrujPocztę niezbędne jest skonfigurowanie w aplikacji katalogu przejściowego skrzynki pocztowej w poleceniu Ustawienia programu -> Parametry pozostałe -> Obieg dokumentów. Standardowo jest to /home/services/finn/skrzpod. Taki katalog należy stworzyć na serwerze i nadać użytkownikowi finn pełne prawa do tego katalogu.

```
# mkdir /home/services/finn/skrzpod
# chown finn.finn /home/services/finn/skrzpod
```

Nazwa funkcji	*Opis*	*Dotyczy*	*Przykład*
cronMrap1	Aktualizuje tablice raportów w generatorze raportów.	wszystkie	`0 * * * * ?###cronMrap1`
cronDaily	Wysyła przez e-mail raporty do użytkowników.	SOD	`0 0 4 * * ?###cronDaily`
rejestrujPoczte	Pobiera i rejestruje pocztę e-mail z kont IMAP/POP3.	SOD	`0 */5 * * * ?###rejestrujPoczte`
sprawdzWiadomosci	Uruchamia procedurę obsługi komunikatów MWD.	SOD, KK, PeUP	`0 * * * * ?###sprawdzWiadomosci`
pformd1Refresh	Wysyła żądanie aktualizacji definicji formularzy.	SOD	`0 0 0 * * ?###pformd1Refresh`
replikujEurzad	Replikuje dane do e-Urzędu FINN.	SOD	`0 0 18 * * ?###replikujEurzad`
cronEcp	Pobiera dane z czytnika czasu pracy Tango. Wymaga konfiguracji czytnika i połączenia kablem szeregowym.	ECP	`0 * * * * ?###cronEcp`
updateCRL	Aktualizuje listy odwołań certyfikatów (CRL) skonfigurowane w poleceniu "Organy certyfikacji". Możliwe jest wywołanie ręczne w w/w poleceniu.	KK, PeUP	`0 */5 * * * ?###updateCRL`
aktualizujUslugi	Uaktualnia wewnętrzne liczniki do szybkiego wyświetlania katalogu usług.	PeUP	`0 */5 * * * ?###aktualizujUslugi`
importSlownikiCbd	Importuje aktualne słowniki z Centralnej Bazy Danych CELAB.	CELAB, LIMS	`0 10 0/6 * * ?###importSlownikiCbd`
exportProbki	Eksportuje wyniki badań do Centralnej Bazy Danych CELAB.	CELAB, LIMS	`0 20 0 * * ?###exportProbki`

1.14. Aktualizacja pakietów oprogramowania wg repozytoriów

Większość oprogramowania systemowego, narzędziowego i aplikacyjnego jest dystrybuowana w postaci pakietów (*ang. package*) RPM. Korzystanie z repozytoriów pakietów umożliwia automatyczne pobieranie i instalowanie pakietów. Repozytoria umożliwiają również automatyczną aktualizację pakietów które zostały udostępnione w nowszej wersji.

1. Sprawdzamy, czy posiadamy wykonaną kopię bezpieczeństwa aktualizowanego systemu.
2. Informujemy odpowiednie struktury o planowanej aktualizacji i ewentualnych przerwach w dostępie do usług.
3. Wyłączamy usługi dostępne dla użytkowników (drobne aktualizacje zwykle nie wymagają przerywania pracy, jednak w przypadku wątpliwości zalecane jest zatrzymanie usług).
4. Aktualizujemy spisy pakietów i wyświetlamy listę pakietów do zaktualizowania (testując jednocześnie prawidłowość pracy repozytoriów).

   ```
   # yum check-update
   ```

   Dostęp do repozytoriów standardowo jest realizowany przez protokół HTTP (port TCP 80). Szczegółów dotyczących konfigurowania repozytoriów należy szukać w dokumentacji programu **yum**. Jeżeli dostęp do Internetu/repozytorium jest możliwy tylko przez serwer proxy (np. Serwery SOD w SEKAP bez dostępu do internetu) to należy ustawić prawidłowo zmienną środowiskową **http_proxy**.
5. Aktualizujemy pakiety.

   ```
   # yum upgrade
   ```

   Program yum ustali pakiety, które wymagają aktualizacji, sprawdzi ich zależności oraz wypisze ich listę. Jeżeli chcemy zaktualizować tylko wybrane pakiety to w poleceniu specyfikujemy ich nazwy (możemy również użyć znaku *, przy czym konieczne może być wtedy zastosowanie cudzysłowów):

   ```
   # yum upgrade rsync "finn-*"
   ```

   Innym przydatnym parametrem jest wykluczenie dane pakietu z aktualizacji. Np. jeśli chcemy zaktualizować cały system, ale bez jądra systemowego, środowiska Java i serwera aplikacji JBoss to korzystamy z polecenia:

   ```
   # yum upgrade -x "kernel*" -x "java*" -x "finn-jboss*"
   ```
6. Akceptujemy wyświetloną listę pakietów. Rozpoczyna się proces pobierania i instalowania pakietów.
7. Po skończeniu aktualizacji zalecany jest restart systemu. Restart systemu jest o tyle korzystny, że automatycznie uruchomi wcześniej zatrzymane usługi. Jeżeli czynności wykonujemy z poziomu zdalnego terminala (brak bezpośredniego dostępu do klawiatury i możliwości wciśnięcia **CTRL-ALT-DEL**) to restart wykonujemy poleceniem:

   ```
   # reboot
   ```

   Uwaga! Wszystkie poważniejsze aktualizacje (w szczególności aktualizacje jądra systemu, *ang. kernel*) należy wykonywać mając zapewniony bezpośredni dostęp do aktualizowanej maszyny.

1.15. Aktualizacja aplikacji FINN

Procedura aktualizacji aplikacji FINN wykorzystuje skrypty narzędziowe z pakietu **finn-common** i przed aktualizacją aplikacji należy zweryfikować ich aktualność. Aplikacja wymaga również odpowiedniej wersji serwera bazy danych PostgreSQL (pakiety **postgres***) oraz serwera aplikacji JBoss (pakiet **finn-jboss4**).

Wyżej wymienione pakiety znajdują się w repozytoriach i można je aktualizować przy pomocy procedury **Aktualizacja pakietów oprogramowania wg repozytoriów**. Wyżej wymieniona procedura zawiera również inne czynności organizacyjne, które powinny być wykonane przed aktualizacją aplikacji FINN.

Aplikację aktualizujemy ZAWSZE o jedną wersję w górę. Nie ma już możliwości aktualizacji o kilka wersji za pomocą baz wzorcowych.

Procedura aktualizacji aplikacji Finn:

1. Sprawdź spis wersji dostępnych w repozytorium.

   ```
   # finn-web-update -l
   ```
2. Dostęp do repozytorium jest realizowany przez protokół RSYNC (port TCP 873). Szczegółów dotyczących konfigurowania klienta należy szukać w dokumentacji programu **rsync**. Jeżeli dostęp do Internetu (repozytorium) jest możliwy tylko przez serwer proxy to należy ustawić prawidłowo zmienną środowiskową **RSYNC_PROXY**.
3. Sprawdź listę baz danych i wersję struktur na jakich są oparte.

   ```
   # finn-db-info
   ```
4. Pobieramy aplikację o jedną wersję wyższą od posiadanej i dokonujemy aktualizacji, np:

   ```
   # finn-web-upgrade -s -w 8.1.11.s.28043 -b
   ```

   Uwaga! W repozytorium są również publikowane wersje testowe i rozwojowe.
5. Instalujemy aplikację w środowisku aplikacyjnym JBoss.

   ```
   # finn-web-install -f
   ```
6. Uruchamiamy serwer aplikacyjny JBoss.

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

7. 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-migrate -h
```

Uwaga! Użytkownicy którzy mają wydzielony serwer bazodanowy powinni również przy poleceniach bazodanowych skorzystać z parametru **-p** aby uniknąć wielokrotnego (około 30 razy) podawania hasła do serwera SQL.

1.16. Konfiguracja automatycznej aktualizacji kursów e-learning

1. Aktualizacja kursów jest oparta o skrypty narzędziowe z pakietu **finn-common**.
2. Weryfikujemy plik konfiguracyjny **/etc/finn/finn.conf** w zakresie repozytorium kursów.
3. Ręcznie uruchamiamy procedurę synchronizacji kursów z repozytorium.

```
# finn-kursy-update -p
```

4. Najprostszą metodą jest wykorzystanie demonu CRON do automatycznego wykonywania skryptu aktualizacyjnego. W tym celu tworzymy odpowiedni link symboliczny do skryptu.

```
# ln -s /usr/sbin/finn-kursy-cron /etc/cron.daily/finn-kursy-cron
```

1.17. Weryfikacja parametrów bezpieczeństwa aplikacji FINN

1. Parametry bezpieczeństwa należy kontrolować uwzględniając wdrożone w podmiocie politykę bezpieczeństwa i procedury pracy.
2. Należy zalogować się do aplikacji FINN użytkownikiem posiadającym wymagane uprawnienia.
3. Otwieramy polecenie **Ustawienia programu** / **Parametry pozostałe** / **Ogólnosystemowe**.
4. Weryfikujemy ustawienia parametrów w grupie **Autoryzacja (logowanie)**:
   a) Długość czasu trwania sesji (w minutach): **10** (zalecany dla podwyższenia poziomu bezpieczeństwa). Parametr może przyjmować wyższą wartość jeżeli stosujemy również inne mechanizmy zabezpieczające stacje robocze, np. automatyczny wygaszacz ekranu blokowany hasłem uruchamiany po 10 minutach.
   b) Kontrola hasła użytkownika na poziomie bezpieczeństwa: **podwyższony/wysoki** (wymagany prawnie, np. ze względu na przetwarzanie danych osobowych).
   c) Minimalna długość hasła: **8** (wymagany prawnie, np. ze względu na przetwarzanie danych osobowych).
   d) Wyłączenie pamiętania hasła w formularzu logowania: **tak** (zalecany dla podwyższenia poziomu bezpieczeństwa).
   e) Liczba niepoprawnych prób logowania: **6** (zalecany dla podwyższenia poziomu bezpieczeństwa).
5. Weryfikujemy ustawienia parametrów w grupie **Historia operacji**:
   a) Pełna rejestracja wykonanych operacji: **tak** (wymagany prawnie, np. ze względu na przetwarzanie danych osobowych).
6. W przypadku modyfikacji parametrów należy wydrukować aktualny ich stan poleceniem **Drukuj** / **Zestawienie parametrów**.

1.18. Awaryjne odzyskanie hasła i uaktywnienie użytkownika oprogramowania FINN

1. Logujemy się na serwer na którym możemy zalogować się do bazy danych. Można to wykonać z bezpośrednio z konsoli serwera lub przy pomocy polecenia PuTTY.
2. Logujemy się do bazy danych. Ewentualnie wcześniej możemy sprawdzić listę dostępnych baz danych na serwerze.

```
$ psql -U postgres -l
            List of databases
      Name        |  Owner   | Encoding
------------------+----------+----------
(tu wypiszą się dostępne bazy danych)
$ psql -U postgres nazwa_bazy_danych
```

3. Wydajemy komendę SQL, która uaktywni użytkownika i nada mu hasło tymczasowe.

```
nazwa_bazy_danych=# UPDATE susr1_user SET user_status = 1, user_password =
'HasloTym123' WHERE user_name = 'finn';
```

Uwaga! Powyższa komenda zapisuje hasło w bazie danych w postaci jawnej. Ponadto powyższa komenda może zostać zapisana w historii poleceń SQL. Dlatego należy niezwłocznie zmienić hasło na docelowe przez interfejs oprogramowania FINN.

2. Archiwizacja

2.1. Konfiguracja systemu archiwizacji autobackup

Do archiwizowania plików i baz danych SQL możemy wykorzystać skrypty z pakietu **finn-admin**. Konfigurację kopii zapasowych znajduje się w pliku **/etc/sysconfig/autobackup**. Aby włączyć automatyczne (raz dziennie) wykonywanie kopii bezpieczeństwa należy w w/w pliku ustawić np.:

```
FILE_PATHS="/etc"
FILE_CRON=yes
PGSQL_CRON=yes
```

W zależności od potrzeb włączamy lub wyłączamy automatyczne wysyłanie kopii do tzw. zdalnej lokalizacji archiwum. W tym celu ustawiamy odpowiednio parametry ***(typ_archiwum)*_SEND** na wartość **yes** lub **no**.

Należy zapoznać się również z pozostałymi parametrami, których opisy znajdują się w pliku.

Uwaga! Skrypty **autobackup** archiwizują tylko najważniejsze dane i nie tworzą kompletnego obrazu potrzebnego do automatycznego odzyskania systemu. Nie mniej tworzą małe i wygodne pliki archiwalne oraz stanowią wartościowe uzupełnienie innych procedur archiwizacji danych.

Aby wymusić natychmiastowe wykonanie kopii, wykonujemy komendę:

```
# /etc/cron.daily/autobackup-cron
```

Aby sprawdzić, ile miejsca zajmuje katalog z paczkami archiwalnymi wykonujemy:

```
# du -sch /opt/backup/*
```

2.2. Archiwizacja baz danych PostgreSQL skryptem autobackup-pgsql

Do archiwizacji baz danych możemy wykorzystać skrypt autobackup-pgsql, który zgodnie z konfiguracją (**/etc/sysconfig/autobackup**) będzie wykonywał kopie zapasowe wszystkich baz danych do plików ***.backup** (standardowo w katalogu **/opt/backup/pgsql**).

```
# autobackup-pgsql
```

2.3. Archiwizacja wybranej bazy danych PostgreSQL poleceniem pg_dump

Wykonujemy polecenie archiwizacji.

```
# pg_dump -F c -f nazwa_bazy.backup nazwa_bazy
```

Szczegółowy opis parametrów polecenia dostępny w dokumentacji PostgreSQL oraz po wydaniu polecenia:

```
# pg_dump –help
```

2.4. 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_set();
   postgres=# \q
   ```
   Weryfikacja sekwencji jest szczególnie istotna jeśli ingerujemy w bazę danych i ręcznie importujemy dane do tabel.

2.5. Archiwizacja najważniejszych plików skryptem autobackup-file

Do archiwizacji najważniejszych plików konfiguracyjnych możemy wykorzystać skrypt autobackup-file, który zgodnie z konfiguracją (**/etc/sysconfig/autobackup**) będzie wykonywał kopię zapasową najważniejszych plików (standardowo jest to katalog **/etc**) w skompresowanym formacie ***.tar.bz2** (standardowo w katalogu **/opt/backup/file**).

```
# autobackup-file
```

2.6. Archiwizacja na płytę DVD, DVDRW

1. Nagrywanie na płytę DVDRW odbywa się za pomocą pakietu **dvd+rw-tools**. Przyjmując, że napędem optycznym jest /dev/hda sprawdzamy najpierw czy w napędzie jest płyta DVD-RW czy DVD+RW:
   ```
   # dvd+rw-mediainfo /dev/hda
   ```
2. Po czym wymazujemy płytę odpowiednim dla formatu płyty poleceniem.
   a) Dla DVD-RW:
      ```
      # dvd+rw-format -force=full -blank=full /dev/hda
      ```
   b) Dla DVD+RW:
      ```
      # dvd+rw-format -force /dev/hda
      ```
3. Nagrywanie odbywa się za pomocą polecenia:
   ```
   # growisofs -Z /dev/hda -R -J -dvd-compat -use-the-force-luke
   /katalog_do_nagrania/
   ```
4. Przed nagraniem płyty DVD (pojemność około 4,3GB) spróbujemy ocenić, ile miejsca zajmuje najbardziej aktualna (dzisiejsza) kopia zapasowa (w katalogu /opt/backup):
   ```
   # du -sch $( find /opt/backup/ -iname '*'$( date '+%Y%m%d' )'*' )
   ```
   Można również określić datę ręcznie (przykład dla 11 sierpnia 2008 roku):
   ```
   # du -sch $( find /opt/backup/ -iname '*20080811*' )
   ```
5. Kolejnym krokiem, który można podjąć aby zmieścić dane na płycie DVD, to **kompresja plików**. Aby skompresować kopie bazy danych z danego dnia (przykład dla 11 sierpnia 2008 roku) wykonujemy:
   ```
   # find /opt/backup/ -iname '*20080811*'  -exec gzip --best {} \;
   ```
   W powyższym przykładzie, użyto programu **find** do wyszczególnienia plików przeznaczonych do kompresji oraz programu **gzip** z parametrem **--best** oznaczającym najlepszą możliwą kompresję.

   Oczywiście lepiej nie kompresować archiwum jeżeli nie jest to konieczne. Odzyskiwanie danych nieskompresowanych z uszkodzonych nośników jest dużo łatwiejsze.
6. Po dokonaniu kompresji, sprawdzamy ponownie, ile zajmuje kopia z danego dnia:
   ```
   # du -sch $( find /opt/backup/ -iname '*20080811*' )
   ```
7. Jeśli nasze działania przyniosły oczekiwany rezultat (pliki mieszczą się na płycie DVD-R), nagrywamy dane na płycie poleceniem:
   ```
   # growisofs -Z /dev/scd0 -R -J -dvd-compat $( find /opt/backup/ -iname
   '*20080811*' )
   ```
8. Gdyby jednak dane z jednego dnia dalej nie mieściły się na płycie DVD-R, niezbędne będzie wykonanie archiwum TAR podzielonego na kawałki o rozmiarach 4,3GB każdy.
9. Upewnijmy się że znajdujemy się w katalogu na partycji, która ma dostatecznie dużo wolnego miejsca na wykonanie podzielonego archiwum tar:
   ```
   cd /opt
   ```
10. Tworzymy wieloczęściowe archiwum tar:
    ```
    # tar -cf - $( find /opt/backup/ -iname '*20080725*' ) | split -b 4250m
    -a4 -d backup-20080725.tar
    ```
    Polecenie to stworzy nam kilka plików ponumerowanych od 0, np:
    ```
    backup-20080725.tar0001
    backup-20080725.tar0002
    backup-20080725.tar0003
    ```
11. Następnie każdy z plików nagrywamy na płyty DVD-R komendą:
    ```
    # growisofs -Z /dev/scd0 -R -J -dvd-compat /opt/backup-20080725.tarNNNN
    ```

Tak nagrane płyty DVD-R należy starannie opisać, najlepiej pełną nazwą pliku, oraz umieścić w bezpiecznym miejscu.

Po pomyślnym nagraniu danych, powinno sie skasować stworzone pliki **backup-20080725.tarNNNN**, aby nie zajmowały niepotrzebnie miejsca na dysku.

Więcej o tych poleceniach w dokumentacji:

```
# man growisofs
# man dvd+rw-format
# man dvd+rw-mediainfo
# man tar
```

Warto również zapoznać się z fragmentami tzw. „Instrukcji kancelarnyj" w zakresie przepisów dotyczących archiwzowania danych (harmonogram wykonywania kopii oraz miejsca ich przechowywania).

2.7. Archiwizacja na taśmę magnetyczną (streamer)

Jeżeli w komputerze jest zainstalowany napęd taśmowy to możliwe jest nagrywanie kopii zapasowych na taśmy magnetyczne. W systemie urządzenie widziane jest jako:

1. /dev/st0 – po nagraniu przewija taśmę do początku, archiwa się nadgrywają,
2. /dev/nst0 – po nagraniu nie przewija taśmy do początku, archiwa są dogrywane.

Streamer można obsługiwać przez polecenia **dump** i **restore**. Na przykład:

1. nagranie katalogu /opt na taśmę streamera po czym przewinie jej do początku:
   ```
   # dump 0af /dev/st0 /opt
   ```
2. nagranie katalogu /opt ale po nagraniu bez przewijania taśmy; kolejne archiwa będą dogrywane:
   ```
   # dump 0af /dev/nst0 /opt
   ```
3. odzyskiwanie danych w trybie interaktywnym, co pozwala na odzyskiwanie poszczególnych katalogów, można używać poleceń dir, cd, extract:
   ```
   # restore ivf /dev/st0
   ```
4. odzyskiwanie całego archiwum do katalogu w którym się aktualnie znajdujemy:
   ```
   # restore xf /dev/st0
   ```

2.8. Kontrola spójności i optymalizacja bazy danych

1. Monitorowanie wielkości bazy danych

   Wielkość baz danych można kontrolować za pomocą kilku poleceń:

   Polecenie df pozwala na sprawdzenie ilości dostępnego miejsca na partycji z bazami danych. Zakładamy oczywiście, że serwer instalowany był zgodnie z DOK.PROC i w systemie jest oddzielna partycja na system baz danych. Przykład:

   ```
   [root@serwer pgsql]# df | grep pgsql
   /dev/sda6             51606140  41267528   7717172  85% /var/lib/pgsql
   ```

   Polecenie du pozwala na sprawdzenie wielkości danego katalogu. Sprawdzamy ile zajmuje katalog /var/lib/pgsql:

   ```
   [root@tpnets-db2 pgsql]# du -s /var/lib/pgsql
   41066884        /var/lib/pgsql
   ```

   Wielkość bazy danych można sprawdzić również za pomocą programu pgAdamin. Po włączeniu programu logujemy się do serwera baz danych i klikamy prawym przyciskiem myszy na pozycji drzewka Bazy danych i wybieramy **Raporty → Raport statystyk**. W polu Plik wyjściowy należy podać nazwę pliku do którego nasz raport będzie generowany i kliknąć OK. Raport zostanie wygenerowany w przyjaznej dla użytkownika formie pliku HTML.

2. Optymalizacja bazy danych

   Optymalizację bazy danych wykonujemy za pomocą polecenia vacuumdb. Przed wykonaniem polecenia zaleca się wyłączenie na serwerze oprogramowania korzystającego z serwera baz danych (programy FK, aplikacja finn-jboss itp.)

   Pełną optymalizację baz wykonujemy poprzez wpisanie w konsoli serwera polecenia:

   ```
   # vacuumdb -a -f -z
   ```

   Można również zoptymalizować wybraną bazę danych:

   ```
   # vacuumdb -d nazwa_bazy -f -z
   ```

3. Więcej na ten temat w dokumentacji:
   ```
   # man vacuumdb
   # vacuumdb –help
   ```
4. Skrypt **finn-db-vacuum-cron**

   W pakiecie **finn-common** w wersji 1.5 znajduje się skrypt finn-db-vacuum-cron dzięki, któremu wykonamy porządkowanie wszystkich baz danych serwera PostgreSQL.

   UWAGA! Skrypt wyłącza serwer aplikacji finn-jboss dlatego nie zaleca się wykonywania go podczas pracy urzędu na aplikacji.

   Aby utrzymać bazę wyporządkowaną i zoptymalizowaną zaleca się dodanie w/w skryptu do cron.weekly. Można to wykonać za pomocą programu mc lub z linii poleceń:
   ```
   # ln -s /usr/sbin/finn-db-vacuum-cron /etc/cron.weekly/finn-db-vacuum-cron
   ```

3. Konserwacja systemu operacyjnego

Poniżej zostaną przedstawione okresowe zabiegi konserwacyjne systemu operacyjnego i bazy danych PostgreSQL94

3.1. Katalogi systemowe

Wolne miejsce sprawdzamy za pomocą polecenia df. W zależności od przeznaczenia serwera musimy okresowo sprawdzać ilość wolnego miejsca w następujących katalogach:

```
/home ─┐
       /services/finn/broker (katalogi output1,output2,output3,output4,input,discarded)
       /services/finn/archiwum
       /services/finn/program
/var/log/finn/jboss-finn
/var/lib/pgsql/wersja_postgresql/data/pg_log
```

3.2. Komunikaty brokera integracyjnego w bazie danych

W tabeli pwiad1 bazy danych SOD odkładają się komunikaty brokera integracyjnego (komunikacja PeUP, SEKAP, e-Urząd) Zakończone komunikaty można skasować, zwalniając tym samym miejsce na dysku.

a) Sprawdzamy ile jest poprawnie zakończonych komunikatów w bazie danych
   ```
   # select count(*) from pwiad1 where stan<>0;
   ```
b) Kasujemy poprawnie zakończone komunikaty
   ```
   # delete from pwiad1 where stan<>0;
   ```

3.3. Konfiguracja automatycznej aktualizacji czasu na serwerze

Jeśli serwer instalowany był zgodnie z F8WEB.PROC to mamy już zainstalowany serwer czasu ntp. Pozostaje jedynie skonfigurować go tak żeby uruchamiał się przy starcie systemu. W tym celu wykonujemy z poziomu użytkownika root polecenia:

Uruchamianie serwera czasu przy starcie systemu:
```
# systemctl enable ntpd
```

Aktualizacja czasu zawsze przy starcie systemu:
```
# systemctl enable ntpdate
```

Datę na serwerze można również zaktualizować ręcznie z dowolnego serwera czasu dostępnego w internecie.

W tym celu wykonujemy polecenie (serwer czasu jest przykładowy):
```
# ntpdate -s time.task.gda.pl
```

4. Konfiguracja sprzętu sieciowego Cisco

Poniższe przykłady są przeznaczone dla routerów Cisco z wersją oprogramowania IOS w wersji 12.4. Przed zastosowaniem ich we własnej konfiguracji należy zapoznać się z dokumentacją techniczną odpowiedniej dla posiadanej wersji systemu IOS i posiadanego sprzętu.

4.1. Synchronizacja czasu na routerze

```
clock timezone CET 1
clock summer-time CET recurring

ntp update-calendar
ntp server 213.222.193.35
ntp server 193.219.28.149
ntp server 150.254.183.15
ntp server 198.123.30.132
```

4.2. Logowanie zdarzeń na zewnętrznym serwerze syslog

```
service timestamps debug datetime msec localtime show-timezone year
service timestamps log datetime msec localtime show-timezone year
service sequence-numbers

logging count
logging buffered 256000 debugging
logging console critical

logging trap debugging
logging adresIPserweraSysLog
```

4.3. Ostrzeżenie przed zalogowaniem

```
banner login .
=== WARNING ===
1. Unauthorized access to this system is prohibited.
2. Use of this system is limited to authorized individuals only.
3. All activities are monitored and logged.
4. There is no privacy on this system.
5. Unauthorized access and activities or any criminal activity
will be reported to appropriate authorities.

=== OSTRZEZENIE ===
1. Nieautoryzowany dostep do tego systemu jest zakazany.
2. Uzywanie tego systemu jest ograniczone tylko do osob upowaznionych.
3. Wszelka aktywnosc jest monitorowana i rejestrowana.
4. W trakcie korzystania nie nalezy oczekiwac zachowania prywatnosci.
5. Nieautoryzowany dostep i wszelkie dzialania niezgodne z prawem beda
zglaszane do odpowiednich organow.

.
```

4.4. Dostęp do routera tylko przez szyfrowany protokół SSH

```
service password-encryption
```

```
security authentication failure rate 3 log
security passwords min-length 8

ip ssh authentication-retries 2

line con 0
 exec-timeout 30 0
 logging synchronous
 transport preferred ssh
 transport output telnet ssh

line vty 0 4
 exec-timeout 30 0
 logging synchronous
 transport preferred ssh
 transport input ssh
 transport output telnet ssh
```

Uwaga! Numery linii mogą zależeć od konkretnej konfiguracji sprzętowej.

4.5. Skrócone komendy

```
alias exec ct configure terminal
```

4.6. Zablokowanie zazwyczaj niepotrzebnych protokołów, przyspieszenie wydajności

```
no service pad
service tcp-keepalives-in
service tcp-keepalives-out

interface Null0
 no ip unreachables

interface wybrany-interfejs
 no ip redirects
 no ip proxy-arp
 no cdp enable
 no mop enabled

no ip forward-protocol udp bootps
no ip forward-protocol udp tftp
no ip forward-protocol udp domain
no ip forward-protocol udp time
no ip forward-protocol udp netbios-ns
no ip forward-protocol udp netbios-dgm
no ip forward-protocol udp tacacs

no ip http server

no cdp run

ip cef
```

```
ip domain retry 3

access-list compiled
```

* KONIEC *