

1. Wstęp
 - Classic Server
 - Super Server
 - Super Classic
2. Firebird
 - Instalacja z repozytorium
 - Ręczna instalacja firebirda
3. Instalacja SAMBY:
4. Apache + PHP
 - SSL
 - vHost
 - Konfiguracja PHP
5. Czyścimy konfigurację
6. Nginx + PHP
 - SSL
 - vHost
 - Konfiguracja PHP



debian

1. Wstęp

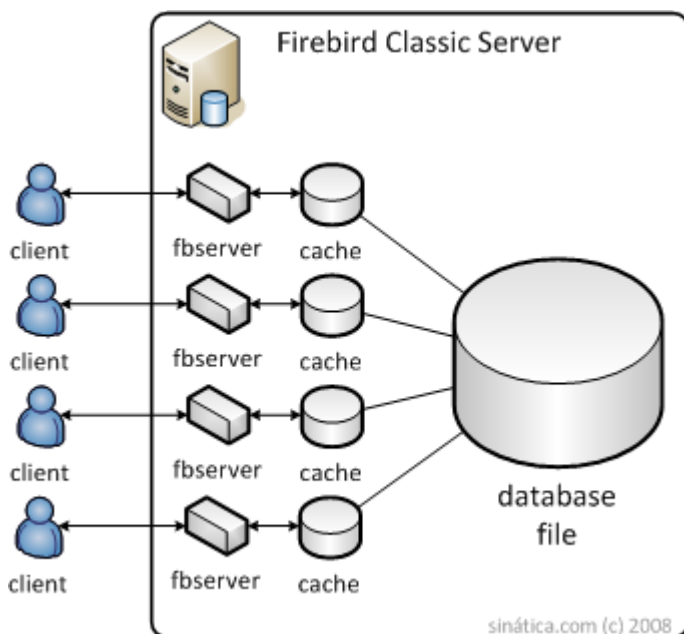
Firebird 2.5 dostępny jest w trzech różnych architekturach:

- Classic Server
- Super Server
- Super Classic

Classic Server

Tworzy proces dla każdego połączenia klienta, każde połączenie ma własną pamięcią podręczną. Wykorzystuje mniej zasobów, jeśli liczba połączeń jest niska. Obecnie Classic jest zalecany do używania w komputerach wieloprocesorowych. Obsługa SMP (wieloprocesorowy).

Classic server installs a `/etc/xinetd.d/firebird` start script or, for older inetd systems, adds an entry to the `/etc/inetd` file

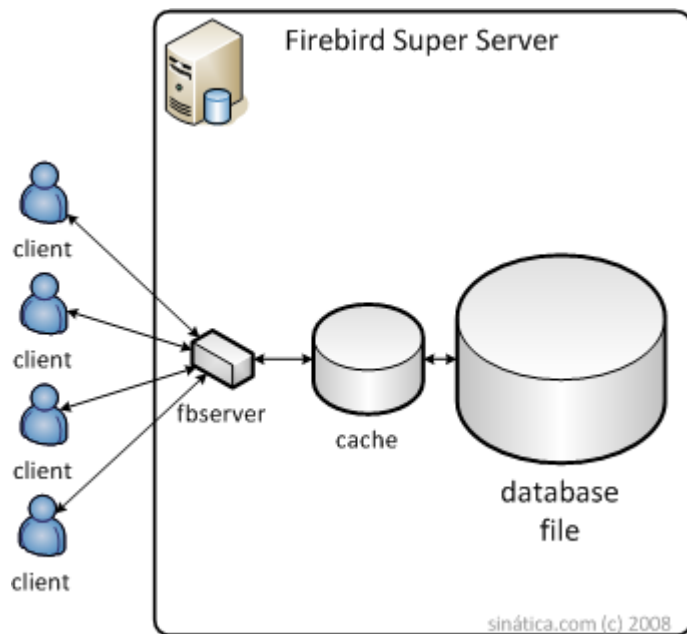


Super Server

Dzieli dostępną pamięć pomiędzy połączenia z bazą danych oraz używa wątków do obsługi każdej transakcji. Staje się bardziej wydajny, jeśli liczba jednoczesnych połączeń jest mała.

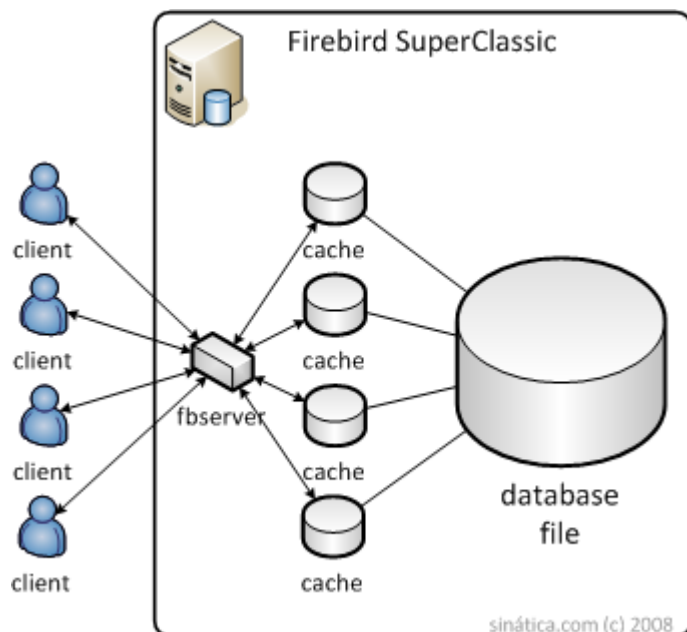
Brak wsparcia dla SMP. Na wieloprocesorowych komputerach z systemem Windows wydajność może znacznie spaść, nawet jeśli system operacyjny przełącza proces między procesorami. Aby temu zapobiec, ustaw parametr `CpuAffinityMask` w pliku konfiguracyjnym `firebird.conf`.

SuperServer only installs a `/etc/rc.d/init.d/firebird` server start script.



Super Classic

Każdy klient ma dedykowany wątek w ramach jednego procesu.
Obsługuje SMP (wieloprocessorowy). Dobra wydajność w każdej sytuacji.



2. Firebird

Instalacja z repozytorium

```
su -  
apt update  
apt upgrade  
apt-cache search Firebird  
apt search Firebird  
apt install firebird3.0-server  
systemctl start firebird3.0  
systemctl enable firebird3.0
```

sprawdzenie:

```
systemctl status firebird3.0  
netstat -tln
```

rekonfiguracja:

```
nano /etc/firebird/3.0/firebird.conf  
systemctl restart firebird3.0  
netstat -tln
```

Firewall

```
iptables -L -n  
ufw status verbose  
ufw allow 3050/tcp
```

Odinstalowanie:

```
apt purge firebird3.0-*
```

Ręczna instalacja firebirda

Pobieramy odpowiednią paczkę ze strony <https://firebirdsql.org/>

```
cd /tensoft/test
```

```
wget https://github.com/FirebirdSQL/firebird/releases/download/R2_5_8/FirebirdCS-2.5.8.27089-0.amd64.tar.gz
```

Rozpakowujemy i instalujemy

```
tar -zxvf FirebirdCS-2.5.8.27089-0.amd64.tar.gz
cd FirebirdCS-2.5.8.27089-0.amd64
apt install xinetd
sh install.sh
```

podajemy hasło dla SYSDBA

Sprawdzenie:

```
netstat -tln
cd /opt/firebird/bin/
ls -al
./isql

connect "localhost:/opt/firebird/examples/empbuild/employee.fdb" user 'SYSDBA' password 'masterkey';
show tables;
exit;
```

testujemy połączenie za pomocą IBExpert lub FlameRobin

Firewall

```
iptables -L -n
ufw status verbose
ufw allow 3050/tcp
```

Biblioteka UDF:

```
find / -name UDF
cp /tensoft/test/udf/Linux_x64/bx_udf.so /opt/firebird/UDF/
chown root /opt/firebird/UDF/bx_udf.so
```

3. Instalacja SAMBY:

```
apt install samba smbclient
systemctl status smbd
```

Konfiguracja:

```
echo "" > /etc/samba/smb.conf  
nano /etc/samba/smb.conf
```

i wklejamy:

```
[global]  
server string = Samba Server %v  
workgroup = UM  
log file = /var/log/samba/log.%m  
max log size = 1000  
syslog = 0  
map to guest = Bad User  
security = USER  
dns proxy = No  
idmap config * : backend = tdb  
#Windows XP Fix  
#server max protocol = NT1  
  
[adas]  
comment = Tensoft Adas  
path = /tensoft/adas  
create mask = 0660  
directory mask = 0770  
read only = No  
force user = tensoft  
force group = tensoft  
valid users = @tensoft
```

Sprawdzamy poprawność konfiguracji:

```
testparm
```

Uprawnienia:

```
mkdir /tensoft/baza  
chown firebird:tensoft /tensoft -R  
chmod 770 /tensoft/ -R  
chmod 660 /tensoft/baza/BXORG.GDB
```

Dodajemy użytkownika dla katalogów:

```
(useradd tensoft -m -G tensoft) ← nie robimy  
(passwd tensoft) ← nie robimy
```

```
smbpasswd -a tensoft
```

Restart

```
systemctl restart nmbd.service  
systemctl restart smb.service  
systemctl enable smb.service  
systemctl enable nmbd.service  
systemctl status smb.service
```

Test:

```
tensoft@debian:~$ smbclient -L localhost -U tensoft
```

Robimy test z komputera windows.

Firewall:

```
ufw status  
ufw allow Samba
```

Połączenia do servera:

```
smbstatus -b
```

(widzimy na jakim protokole łączą się klienci)

4. Apache + PHP

Dodajemy dodatkowe repozytorium dla PHP

```
wget -q https://packages.sury.org/php/apt.gpg -O- | apt-key add -  
echo "deb https://packages.sury.org/php/ stretch main" | tee /etc/apt/sources.list.d/php.list  
apt-get install ca-certificates apt-transport-https  
apt update
```

Sprawdzamy dostępne wersje php

```
apt-cache search php
```

Instalujemy:

```
apt install apache2 libapache2-mod-php7.2
```

Sprawdzamy czy się zainstalował i uruchomił

```
systemctl status apache2  
php -v
```

sprawdzamy w przeglądarce czy działa domyślna strona Apache2

Firewall

```
ufw status  
ufw allow 80  
ufw allow 8787
```

SSL

Generujemy certyfikaty SSL (ważne 5 lat) potrzebne do szyfrowanego połączenia

```
mkdir /etc/apache2/ssl  
openssl req -x509 -nodes -days 1825 -newkey rsa:2048 -keyout /etc/apache  
2/ssl/apache.key -out /etc/apache2/ssl/apache.crt
```

Podajemy oczywiście dane urzędu a nie Tensoftu:

```
Country Name (2 letter code) [AU]:PL  
State or Province Name (full name) [Some-State]:opolskie  
Locality Name (eg, city) []:Opole  
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Tensoft Sp. z  
o.o.  
Organizational Unit Name (eg, section) []:eVAT  
Common Name (e.g. server FQDN or YOUR name) []:www.evat.tensoft.pl  
Email Address []:admin@tensoft.pl
```

vHost

Tworzymy plik konfiguracyjny dla eVAT. W tym przykładzie Apache2 będzie działał na porcie 8787

```
nano /etc/apache2/sites-available/evat.conf
```


o zawartości

```
<IfModule mod_ssl.c>
<VirtualHost _default_:8787>
ServerAdmin webmaster@localhost
DocumentRoot /var/www/html/evat/web

SSLEngine on
SSLCertificateFile /etc/apache2/ssl/apache.crt
SSLCertificateKeyFile /etc/apache2/ssl/apache.key
<FilesMatch "\.(cgi|shtml|phtml|php)$">
SSLOptions +StdEnvVars
</FilesMatch>
<Directory /usr/lib/cgi-bin>
SSLOptions +StdEnvVars
</Directory>

<Directory /var/www/html/evat/web>
Require all granted
AllowOverride All
Order Allow,Deny
Allow from All
<IfModule mod_rewrite.c>
RewriteEngine On
RewriteCond %{REQUEST_FILENAME} !-f
RewriteRule ^(.*)$ /app_dev.php [QSA,L]
</IfModule>
</Directory>

ErrorLog /var/www/html/evat/evat_error.log
CustomLog /var/www/html/evat/evat_access.log combined
</VirtualHost></IfModule>
```

włączamy konfigurację

```
ln -s /etc/apache2/sites-available/evat.conf /etc/apache2/sites-enabled/
lub
a2ensite evat
```

Dodajemy port na którym Apache będzie działał

```
nano /etc/apache2/ports.conf
```

zawartość:

```
<IfModule ssl_module>
Listen 8787
```

```
</IfModule>

<IfModule mod_gnutls.c>
Listen 8787
</IfModule>
```

Tworzymy katalog ze stroną

```
mkdir -p /var/www/html/evat/web
```

Włączamy konfigurację

```
a2enmod ssl
a2enmod php7.2
a2enmod rewrite
systemctl restart apache2
```

Sprawdzamy:

```
netstat -tpln
echo "<?php phpinfo(); ?>" >/var/www/html/evat/web/app.php
```

wywołujemy w przeglądarce stronę <https://nasznumerip:8787/>

Konfiguracja PHP

Instalujemy dodatkowe moduły:

```
apt install php7.2-fpm php7.2-curl php7.2-interbase php7.2-gd php7.2-xml
php7.2-soap php7.2-curl php7.2-intl php7.2-mbstring
```

Konfigurujemy PHP

```
nano /etc/php/7.2/apache2/php.ini
```

i ustawiamy

```
max_execution_time = 600
max_input_time = 600
memory_limit = 1024M
post_max_size = 100M
upload_max_filesize = 100M
cgi.fix_pathinfo=0
```

```
date.timezone = Europe/Warsaw
```

Restartujemy proces PHP

```
systemctl restart php7.2-fpm.service
```

Restart Apache2

```
systemctl restart apache2
```

5. Czyścimy konfigurację

```
apt purge apache2*  
apt purge php7*
```

6. Nginx + PHP

Dodajemy dodatkowe repozytorium dla PHP

```
wget -q https://packages.sury.org/php/apt.gpg -O- | apt-key add -  
echo "deb https://packages.sury.org/php/ stretch main" | tee /etc/apt/sources.list.d/php.list  
apt-get install ca-certificates apt-transport-https  
apt update
```

Sprawdzamy dostępne wersje php

```
apt-cache search php
```

Instalujemy:

```
apt install nginx php7.2-fpm
```

Sprawdzamy czy się zainstalował i uruchomił

```
systemctl status nginx  
php -v
```

sprawdzamy w przeglądarce czy działa domyślna strona Nginx

Firewall

```
ufw status
ufw allow 80
ufw allow 8787
```

SSL

Generujemy certyfikaty SSL (ważne 5 lat) potrzebne do szyfrowanego połączenia

```
mkdir /etc/nginx/ssl
openssl req -x509 -nodes -days 1825 -newkey rsa:2048 -keyout /etc/nginx/ssl/nginx.key -out /etc/nginx/ssl/nginx.crt
```

Podajemy oczywiście dane urzędu a nie Tensoftu:

```
Country Name (2 letter code) [AU]:PL
State or Province Name (full name) [Some-State]:opolskie
Locality Name (eg, city) []:Opole
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Tensoft Sp. z o.o.
Organizational Unit Name (eg, section) []:eVAT
Common Name (e.g. server FQDN or YOUR name) []:www.evat.tensoft.pl
Email Address []:admin@tensoft.pl
```

vHost

Tworzymy plik konfiguracyjny dla eVAT. W tym przykładzie Nginx będzie działał na porcie 8787

```
nano /etc/nginx/sites-available/evat
```

o zawartości

```
server {
# SSL configuration on port 8787
listen 8787 ssl default_server;
#
ssl_certificate /etc/nginx/ssl/nginx.crt;
ssl_certificate_key /etc/nginx/ssl/nginx.key;
```

```
#katalog z aplikacją evat
root /var/www/html/evat/web;

#tuning
client_max_body_size 300M;
client_header_timeout 3000;
client_body_timeout 3000;
fastcgi_read_timeout 3000;
fastcgi_buffers 8 128k;
fastcgi_buffer_size 128k;
fastcgi_busy_buffers_size 256k;

# Add index.php to the list if you are using PHP
index index.html index.htm app.php index.php;

server_name evat.jakas_domena.pl;

location / {
    try_files $uri /app.php$is_args$args;
}

location ~ ^/(app|app_dev|check|config)\.php(/|$) {
    fastcgi_split_path_info ^(.+\.(php))(/.*)$;
    include /etc/nginx/fastcgi_params;
    fastcgi_pass unix:/var/run/php/php7.2-fpm.sock;
    fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
}

location ~ \.php$ {
    return 404;
}

location ~ /\.ht {
    deny all;
}

#Logi
access_log /var/www/html/evat/evat.access.log;
error_log /var/www/html/evat/evat.error.log error;
}
```

Włączamy konfigurację

```
ln -s /etc/nginx/sites-available/evat /etc/nginx/sites-enabled/evat
```

Tworzymy katalog ze stroną

```
mkdir -p /var/www/html/evat/web
```

Sprawdzamy:

```
systemctl restart nginx
systemctl status nginx
netstat -tln
echo "<?php phpinfo(); ?>" >/var/www/html/evat/web/app.php
```

wywołujemy w przeglądarce stronę <https://nasznumerip:8787/>

Konfiguracja PHP

Instalujemy dodatkowe moduły:

```
apt install php7.2-fpm php7.2-curl php7.2-interbase php7.2-gd php7.2-xml
php7.2-soap php7.2-curl php7.2-intl php7.2-mbstring
```

Konfigurujemy PHP

```
nano /etc/php/7.2/fpm/php.ini
```

i ustawiamy

```
max_execution_time = 600
max_input_time = 600
memory_limit = 1024M
post_max_size = 100M
upload_max_filesize = 100M
cgi.fix_pathinfo=0
date.timezone = Europe/Warsaw
```

Restartujemy proces PHP

```
systemctl restart php7.2-fpm.service
```

Restart Nginx

```
systemctl restart nginx
```