

Aktualizacja systemu Debian 8 Jessie do Debian 9 Stretch

		Debian 8 (Jessie)	Debian 9 (Stretch)
Domyślne	PostgreSQL	9.6	10
	PHP	5.6	7.0
Opcjonalne	PostgreSQL	10	11
	PHP	5.6	downgrade 5.6/ upgrade 7.2

Support dla Debian oraz PHP

Support dla Debian 8 <https://wiki.debian.org/LTS/>

Support dla PHP <http://php.net/supported-versions.php>

3. Downgrade do PHP 5.6 lub aktualizacja do PHP 7.2

Ten punkt zostanie podzielony na dwa oddzielne w zależności od tego jaką ścieżkę wybierzemy przechodzimy do podpunktu a) Downgrade do PHP 5.6 lub podpunktu b) Aktualizacja do PHP 7.2

3a. Downgrade do PHP 5.6

Po aktualizacji systemu do Linux Debian 9, Stretch, na naszym serwerze dostępna będzie wersja PHP 7.0:

```
php -v
```

Aktualizacja nie odinstaluje poprzedniej wersji czyli PHP 5.6 z której będziemy mogli nadal korzystać wystarczy tylko wykonać poniższe polecenie które odinstaluje PHP 7.0:

```
sudo apt-get remove php7.0*
```

Po tej czynności nadal będziemy mogli korzystać z eDokumentów działających na PHP w wersji 5.6 Odtworzenie linku symbolicznego do php.ini

```
ln -s /etc/php5/apache2/php.ini /etc/php5/cli/
```

Przy pozostaniu na wersji php 5.6 - eDokumenty będą działać tak jak do tej pory.

3b. Aktualizacja do PHP 7.2

W tym podpunkcie zajmiemy się aktualizacją PHP do wersji 7.2 a także zmianą ZendLoadera na IONCubeLoadera.

Przed instalacją nowych pakietów musimy posprzątać nie potrzebne elementy w tym PHP oraz ZendLoadera. Jeśli korzystaliśmy z PHP 5.6 wykonujemy poniższe polecenie:

```
apt-get purge php5 php5-common php5-dev php5-imap php5-ldap php5-odbc php5-readline php5-sybase php5-cli php5-curl php5-gd
```

Jeśli świeżo aktualizowaliśmy do Debian 9 Stretch możemy również usunąć pakiety PHP 7.0 poniższym poleceniem.

```
sudo apt-get remove php7.0*
```

Usuńmy również ręcznie pliki ZendLoadera nie będą nam już potrzebne. Pliki powinny znajdować się w lokalizacji:

```
/usr/lib/php5/ZendGuardLoader.so  
/usr/lib/php5/opcache.so
```

Przystępujemy do instalacji PHP 7.2 w środowisku Debian 9. Dodajemy repozytoria dla PHP 7.2

```
sudo apt install ca-certificates apt-transport-https  
wget -q https://packages.sury.org/php/apt.gpg -O- | sudo apt-key add -
```

```
echo "deb https://packages.sury.org/php/ stretch main" | sudo tee /etc/apt/sources.list.d/php.list
```

```
apt-get update
```

Przystępujemy do instalacji pakietów PHP 7.2:

```
apt install php7.2-fpm php7.2-cli php7.2-common php7.2-curl php7.2-mbstring php7.2-mysql php7.2-xml php7.2-gd php7.2-pgsql
```

```
apt-get install php7.2-zip
```

Dla FPM instalujemy REDIS zamiast memcached

```
apt-get install redis-server
apt-get install php7.2-redis
```

Konfiguracja dla Redis, którą wprowadzamy w pliku **/etc/redis/redis.conf**

```
save 1200 1
save 600 10
save 120 10000
```

Włączamy obsługę modułów w Apache

```
sudo a2enmod actions alias proxy_fcgi
```

Dodajemy IonCubeLoadera, plik znajduje się w paczce konfiguracyjnej [tutaj](#). Po umieszczeniu go na serwerze np

```
/usr/lib/php/
```

Dodajemy w pierwszej linii pliku php.ini w lokalizacjach

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

następujący wpis:

```
zend_extension=/usr/lib/php/ioncube_loader_lin_7.2.so
```

W plikach php.ini dodajemy również

```
session.save_handler = redis
session.save_path = "tcp://127.0.0.1:6379"
```

oraz modyfikujemy wartości dla:

```
memory_limit = 512M
upload_max_filesize = 32M
post_max_size = 64M
expose_php = Off
```

Kolejnym krokiem jest modyfikacja pliku www.conf

```
vim /etc/php/7.2/fpm/pool.d/www.conf
```

modyfikujemy wpis

```
security.limit_extensions = .php engine
pm.max_children = 8
```

Ostatnim elementem jest zmiana handlera w Apache w pliku default.conf oraz default-ssl.conf, obecny wpis:

```
<FilesMatch "(.+[.]inc$|.+[.]php$|engine)">
    SetHandler application/x-httpd-php
</FilesMatch>
```

modyfikujemy następująco

```
<FilesMatch "(.+[.]inc$|.+[.]php$|engine$)">
    SetHandler "proxy:unix:/var/run/php/php7.2-fpm.sock|fcgi://localhost/"
</FilesMatch>
```

W plikach konfiguracyjnych default.conf oraz default-ssl.conf Apache dodajemy również, na potrzeby autoryzacji WebDAV oraz CalDAV:

```
SetEnvIf Authorization "(.*)" HTTP_AUTHORIZATION=$1
```

Z poziomu konsoli wykonujemy polecenie:

```
a2enmod headers
```

W plikach konfiguracyjnych **conf-enabled/security.conf** w Apache wprowadzamy następujące zmiany:

```
ServerSignature Off
Header set X-Content-Type-Options: "nosniff"
Header set X-Frame-Options: "sameorigin"
Header always set Referrer-Policy "same-origin"
```

Uruchomienie modułu tłumaczącego dla Ready™. Dodajemy moduł do php 7.2 - bstedt.so w lokalizacji

```
/usr/lib/php/20170718
```

Następnie tworzymy pliki bstedt.ini w folderach php:

```
/etc/php/7.2/mods-available/bstedt.ini
```

Zawartość pliku bstedt.ini

```
extension=bstedt.so
```

Tworzymy symlinki do pliku bstedt.ini

```
ln -s /etc/php/7.2/mods-available/bstedt.ini /etc/php/7.2/cli/conf.d/bstedt.ini
ln -s /etc/php/7.2/mods-available/bstedt.ini /etc/php/7.2/fpm/conf.d/bstedt.ini
```

W pliku **php.ini** dla **/etc/php/7.2/cli/** oraz **/etc/php/7.2/fpm/** dodajemy wpis:

```
[bstedt]
bstedt.lang_path="/home/edokumenty/public_html/apps/edokumenty/share/lang"
bstedt.lang=en_UK
```

Wyłączamy moduł opcache poprzez zakomentowanie w **/etc/php/7.2/mods-available/opcache.ini**

Po zakończeniu konfiguracji restartujemy usługi Apache oraz FPM

```
service php7.2-fpm restart
/etc/init.d/apache2 restart
```

Gotowe pliki konfiguracyjne można pobrać z [poniższej lokalizacji](#). Po wgraniu plików wystarczy tylko restart usług Apache oraz PHP-FPM

Po zmianie PHP na 7.2 konieczna jest aktualizacja kodu system Ready_™ do kompilacji dla php 7.2. Aktualizacja kodu na nowszą wersję może wymagać wykonania patchy. Aktualizację wykonujemy w taki sam sposób jak dla systemu eDokumenty. Pomocny może być w tym przypadku ten [link](#).