

## 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>

Poniższa instrukcja przedstawia sposób aktualizacji systemu operacyjnego Linux. Aktualizacja systemu jest przeprowadzana w celu podniesienia wersji PHP do wersji 7.2, jest także możliwość pozostania przy wersji PHP 5.6 (PHP 5.6 przestaje być wspierany w grudniu 2018r).

Jeśli aktualizujemy system z wersji 7 Debian zalecamy zapoznanie się z instrukcją [Aktualizacja systemu Debian 7 Wheezy do Debian 8 Jessie](#), która zawiera wszelkie uwagi dotyczące zmian w Apache gdzie aktualizowany jest z wersji 2.2 do 2.4.

**Przed przystąpieniem do prac BEZWZGLĘDnie należy pamiętać o wykonaniu KOPI BEZPIECZEŃSTWA, całego systemu, bazy danych, katalogu /home/edokumenty.**

### 1. Aktualizacja systemu w gałęzi Jessie, Debian 8

Zaczynamy od aktualizacji systemu w ramach gałęzi, której aktualnie się znajduje (tj. Jessie, Debian 8):

```
#apt-get update
#apt-get upgrade
```

Jeśli system nie był długo aktualizowany pewnie trzeba będzie uruchomić system ponownie ze względu na nowszą wersję jądra systemu.

```
#reboot
```

Przed aktualizacją dystrybucji wykonujemy kopię klików konfiguracyjnych

```
tar -pczf etc.tar.gz /etc
```

### 2. Aktualizacja dystrybucji z Jessie na Stretch, Debian 8 na Debian 9

Po aktualizacji systemu edytujemy plik /etc/apt/sources.list, zmieniamy dotychczasowe wpisy dla Jessie na Stretch

```
deb http://httpredir.debian.org/debian stretch main
deb http://httpredir.debian.org/debian stretch-updates main
deb http://security.debian.org stretch/updates main
```

Możemy również skorzystać z polecenia, który wykona zmiany

```
sed -i 's/jessie/stretch/g' /etc/apt/sources.list
```

Następnym krokiem jest zaktualizowanie pakietów

```
apt-get update
```

oraz aktualizacja dystrybucji

```
apt-get dist-upgrade
```

W niektórych przypadkach aktualizacji dystrybucji systemu mogą pojawić się problemy z aktualizacją komponentów PHP. W tym przypadku należy usunąć link symboliczny do php.ini z /etc/php5/cli i przeniesienie php.ini z /etc/php5/apache2/. Po zakończonej aktualizacji i restarcie systemu ponownie tworzymy link symboliczny do php.ini. Wznowienie aktualizacji wykonujemy poleceniem

```
apt-get -f install
```

Po zakończonej aktualizacji wykonujemy restart systemu

```
reboot
```

### 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/
```

#### 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-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/php/ZendGuardLoader.so  
/usr/lib/php/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 php-zip
```

Modułu do Apache2 dla FCGI

```
apt install apache2 libapache2-mod-fcgid
```

Dla FCGI 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 modułu FCGI w Apache

```
sudo a2enmod actions fcgid 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 = 0ff
```

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"
```

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

#### 4. Aktualizacja PostgreSQLa do wersji 10/11

Przystępujemy do aktualizacji PostgreSQLa do wersji 10/11. Po zainstalowaniu PostgreSQLa będą dostępne dwa klastry dla wersji 9.6 oraz wersji 10/11, odpowiednio na portach 5432 i 5433. Instalację przeprowadzamy za pomocą polecenia

Aktualizujemy dotyczący pakietów dla nowej wersji Debiana obecnie jest to stretch

```
deb http://apt.postgresql.org/pub/repos/apt/ stretch-pgdg main
apt-get update
```

```
apt-get install postgresql-10
```

lub

```
apt-get install postgresql-11
```

Po zainstalowaniu PostgreSQL a przed odtworzeniem bazy należy pamiętać aby wgrać pliki tsearch\_data dla wyszukiwarki.

```
wget http://support.edokumenty.eu/download/deployment/components/tsearch_data_polish_20120730.zip
unzip tsearch_data_polish_20120730.zip
cp tsearch_data_polish/* /usr/share/postgresql/[WERSJA_POSTGRESQL]/tsearch_data/
```

Po tych czynnościach możemy przystąpić do odtworzenia bazy danych

Możemy wykonać zrzut bazy z PostgreSQL na porcie 5432.

```
#su - postgres
$psql -U edokumenty -f edokumenty_upgrade.sql edokumenty (polecenie zakłada że edokumenty działają na bazie edokumenty i w
```

W nowym klastrze trzeba utworzyć użytkownika oraz bazę:

```
$createuser -p 5433 --no-superuser --no-createdb --no-createrole http
$createuser -p 5433 --no-superuser --no-createdb --no-createrole edokumenty
$createdb -p 5433 -E UTF8 edokumenty -O edokumenty
```

Następnie wrzucamy do serwera psql zrzut wykonany chwilę wcześniej:

```
$psql -p 5433 -U edokumenty -f edokumenty_upgrade.sql edokumenty
```

Następnym krokiem jest zatrzymanie PostgreSQL

```
/etc/init.d/postgresql stop
```

W pliku konfiguracyjnym zmieniamy porty dla PostgreSQL aby nowy działał na domyślnym porcie 5432 Starego PostgreSQL zmieniamy na 5433

```
vim /etc/postgresql/9.6/main/postgresql.conf
port=5433
```

W konfiguracji nowego PostgreSQL zmieniamy na 5432

```
vim /etc/postgresql/10/main/postgresql.conf
port=5432
```

W pliku konfiguracyjnym pg\_hba.conf dla aktualnej wersji PostgreSQL dodajemy wpisy,

```
vim /etc/postgresql/10/main/pg_hba.conf

local edokumenty edokumenty md5
local edokumenty http md5
```

Następnym krokiem jest uruchomienie PostgreSQL

```
/etc/init.d/postgresql start
```

Możemy również wykonać aktualizację klastra za pomocą poleceń opisanych w artykule [Upgrade PostgreSQL](#)

W przypadku aktualizacji z Linux Debian 8 większość elementów została już wykonana przy okazji aktualizacji do tej wersji systemu. W Debian 8 powinniśmy już posiadać zaktualizowany repozytorium SVN opisane w poniższym artykule [Aktualizacja systemu Debian 7 Wheezy do Debian 8 Jessie](#), a także podmieniony Zend Loader dla PHP 5.6, poprawnie prawnie skonfigurowanego Apache a także PostgreSQL a w wersji 9.6