

Instalasi & Konfigurasi Laravel dengan php5.6 di CentOS 7 di Cloudmatika

Catata berikut mendokumentasikan proses migrasi infrastruktur myapps dari existing yang berbasis CT (Container) ke server baru yang berbasis VM (KVM).

- Konfigurasi `resolv.conf`

Nah, karena ada isu koneksi ke repository saat awal-awal melakukan konfigurasi dan perlu menambahkan/mengubah berkas `resolv.conf` agar mengarah ke DNS server milik cloudmatika. Maka gunakan IP berikut untuk ditambahkan:

```
[root@myapps-server-vm ~]# cat /etc/resolv.conf
nameserver 8.8.8.8
nameserver 8.8.4.4

; generated by /usr/sbin/dhclient-script
#nameserver 192.168.0.3
#nameserver 192.168.0.2
```

IP 192.168.x.y merupakan bawaan instalasi diberi tanda # di depan.

1. Instalasi paket yang dibutuhkan

```
yum update
yum install epel-release
rpm -Uvh http://rpms.famillecollet.com/enterprise/remi-release-7.rpm
yum install php56-php-
{devel,fpm,fpm,gd,json,mbstring,mcrypt,mysqlnd,odbc,pdo,process,soap,xml,xml
rpc,cli,pear} redis rsync mariadb-server openvpn tmux httpd wget http vim
zip supervisor
php -v
```

Proses di atas adalah untuk memasang paket-paket yang dibutuhkan. Baris terakhir akan menampilkan error karena path php belum didaftarkan ke sistem.

- Aktifkan php cli

```
cat /opt/remi/php56/enable
. /opt/remi/php56/enable
which php
php -v
```

Perintah-perintah di atas untuk menguji script yang telah disediakan oleh repo remi untuk mengaktifkan path php56. Baris terakhir seharusnya telah menampilkan path php yang berada di `/opt/remi/php56/root/usr/bin/php` yang artinya php cli telah dapat digunakan. Keluarannya kurang lebih seperti berikut:

```
PHP 5.6.40 (cli) (built: Apr 28 2021 14:35:32)
Copyright (c) 1997-2016 The PHP Group
```

Zend Engine v2.6.0, Copyright (c) 1998-2016 Zend Technologies

Untuk mendaftarkan secara default kita akan menempatkan script ke dalam direktori `/etc/profile.d/` agar dapat dibaca oleh seluruh sistem. Buat file `php.sh`

```
vim /etc/profile.d/php.sh
```

lalu letakkan baris berikut:

```
. /opt/remi/php56/enable
```

Coba logout & login lagi, seharusnya sekarang sudah dapat langsung menggunakan php. Untuk melihat path php yang aktif bisa diperiksa dengan perintah:

```
which php
```

- Buat user yang akan digunakan oleh aplikasi

Untuk membuat user, dapat menggunakan perintah berikut:

```
adduser -m -U -s /bin/bash myapps  
su - myapps
```

- Konfigurasi php-fpm

Kembali ke user root, lalu buat konfigurasi php-fpm untuk user myapps

```
cd /etc/opt/remi/php56/php-fpm.d  
cp www.conf myapps.conf
```

Konten konfigurasinya kurang lebih sebagai berikut:

```
[myapps]  
  
user = myapps  
group = myapps  
  
;listen = 127.0.0.1:9000  
listen = /run/php56-fpm.sock  
  
listen.owner = myapps  
listen.group = myapps  
;listen.mode = 0660  
listen.allowed_clients = 127.0.0.1  
  
pm = dynamic  
pm.max_children = 200  
pm.start_servers = 10  
pm.min_spare_servers = 5
```

```
pm.max_spare_servers = 35
;pm.process_idle_timeout = 10s;
;pm.max_requests = 500
;pm.status_path = /status
;ping.path = /ping
;ping.response = pong

;access.log = log/$pool.access.log
;access.format = "%R - %u %t |\"%m %r%Q%q|" %s %f %{mili}d %{kilo}M %C%"
slowlog = /opt/remi/php56/root/var/log/php-fpm/www-slow.log
;request_slowlog_timeout = 0
;request_terminate_timeout = 0
;rlimit_files = 1024
;rlimit_core = 0
;chroot =
;chdir = /var/www
;catch_workers_output = yes
;clear_env = no
;security.limit_extensions = .php .php3 .php4 .php5
;env[HOSTNAME] = $HOSTNAME
;env[PATH] = /usr/local/bin:/usr/bin:/bin
;env[TMP] = /tmp
;env[TMPDIR] = /tmp
;env[TEMP] = /tmp
;php_admin_value[sendmail_path] = /usr/sbin/sendmail -t -i -f
www@my.domain.com
;php_flag[display_errors] = off
php_admin_value[error_log] = /opt/remi/php56/root/var/log/php-fpm/www-
error.log
php_admin_flag[log_errors] = on
;php_admin_value[memory_limit] = 128M
php_value[session.save_handler] = files
php_value[session.save_path] = /opt/remi/php56/root/var/lib/php/session
php_value[soap.wsdl_cache_dir] = /opt/remi/php56/root/var/lib/php/wsdlcache
```

Beberapa penyesuaian antara lain terkait user & path listen file.

```
systemctl status php56-php-fpm
cat /usr/lib/systemd/system/php56-php-fpm.service
```

Aktifkan service php-fpm-nya dengan perintah-perintah berikut:

```
systemctl enable php56-php-fpm
systemctl start php56-php-fpm
systemctl status php56-php-fpm
```

- Instalasi composer

Agar dapat menggunakan composer, kita perlu memasangnya terlebih dahulu. Nah, di sini kita kan meletakkannya di direktori bin dari php56 yang telah kita pasang sebelumnya. Cari tau dulu path-nya ada di mana, lalu kita akan berpindah ke sana:

```
which php
cd /opt/remi/php56/root/usr/bin/
```

Selanjutnya download binary composer, beri akses execute, dan buat symlink menjadi composer:

```
wget https://github.com/composer/composer/releases/download/1.10.21/composer.phar
chmod +x composer.phar
ln -s composer.phar composer
```

Sampe di sini composer telah dapat digunakan.

- Konfigurasi httpd

Yang tak kalah penting adalah konfigurasi httpd. Kita cek apakah service httpd sudah jalan atau belum, lalu coba aktifkan.

```
ss -tulpn
systemctl start httpd
systemctl enable httpd
systemctl status httpd
```

Untuk mencoba langsung dari cli bisa dicoba dengan curl:

```
curl localhost
```

- Buat VirtualHost untuk myapps corporate

bagian konfigurasi, kita hapus konfigurasi default berupa welcome.conf lalu kita buat file baru myapps.conf

```
cd /etc/httpd/conf.d/
rm welcome.conf
vim myapps.conf
```

konten file tersebut kurang lebih sebagai berikut:

```
<VirtualHost *:80 >
    ServerName "myapps.co.id"
    ServerAlias "www.myapps.co.id"
    ServerAlias "apps.myapps.co.id"
    UseCanonicalName Off

    DocumentRoot "/home/myapps/httpdocs"
    CustomLog /home/myapps/logs/access_log combined
    ErrorLog /home/myapps/logs/error_log

    <IfModule mod_suexec.c>
        SuexecUserGroup "myapps" "myapps"
    </IfModule>
```

```
<FilesMatch \.php$>  
    SetHandler "proxy:unix:/run/php56-fpm.sock|fcgi://localhost/"  
</FilesMatch>
```

```
Timeout 120
```

```
<Directory /home/myapps/httpdocs>
```

```
    #Options +Indexes +FollowSymLinks +MultiViews  
    AllowOverride All  
    Order allow,deny  
    allow from all  
    Require all granted
```

```
</Directory>
```

```
#<IfModule mod_rewrite.c>  
# RewriteEngine On  
# RewriteCond %{HTTP_HOST} ^myapps.co.id$ [NC]  
# RewriteRule ^(.*)$ http://www.myapps.co.id$1 [L,R=301]  
#</IfModule>
```

```
</VirtualHost>
```

Tentunya, lakukan modifikasi lebih lanjut jika diperlukan. Pastikan konfigurasi httpd-nya tidak ada yang error dengan mengetikkan:

```
apachectl configtest
```

kalau hasilnya muncul Syntax OK berarti sudah benar konfigurasi-nya. Restart service httpd:

```
systemctl restart httpd  
systemctl status httpd
```

- Konfigurasi mariadb

Jalankan perintah berikut untuk pertama kalinya untuk memberi password pada akun root pada mariadb:

```
mysql_secure_installation
```

Simpan credential user root di home directory-nya user root:

```
vim /root/.my.cnf
```

kontennya sebagai berikut:

```
[client]
```

```
user=root  
password=mysecurepassword
```

Catatan: Ganti mysecurepassword dengan password yang aman, misalnya random karakter.

Oiya, kalo ternyata belum aktif ya diaktifkan dulu:

```
systemctl enable mariadb  
systemctl status mariadb  
systemctl start mariadb
```

Buat database untuk prod:

```
MariaDB [(none)]> create database myapps_production;  
MariaDB [(none)]> grant all privileges on myapps_production.* to  
'myapps'@'localhost' identified by 'myrandompassword';  
MariaDB [(none)]> flush privileges;
```

Impor database dari file backup:

```
cd /home/myapps/migrasi/  
zcat myapps_production.sql.gz | mysql -u root myapps_production
```

Catatan: file backup telah disediakan.

- Pasang ffmpeg, karena ternyata myapps membutuhkannya

Jadi begini cara install-nya:

```
yum localinstall --nogpgcheck  
https://download1.rpmfusion.org/free/el/rpmfusion-free-release-7.noarch.rpm  
yum install ffmpeg ffmpeg-devel
```

Sudah, itu aja.

- Konfigurasi supervisor

Supervisor dibutuhkan oleh myapps:

```
cd /etc/supervisord.d/  
vim myapps.conf
```

Isi konten konfigurasinya kurang lebih begini:

```
[program:myapps]  
command=/opt/remi/php56/root/usr/bin/php artisan queue:work --sleep=1 --  
tries=3  
directory=/home/myapps/httpdocs  
autostart=true
```

```
autorestart=true
user=myapps
startsecs=10
startretries=10
log_stdout=true
log_stderr=true
logfile=/home/myapps/logs/supervisor/myapps.log
logfile_maxbytes=2MB
logfile_backups=7
```

Kalo ternyata ada yang kurang kompatibel, disesuaikan lagi aja ya. Karena khusus yang ini belum dites.

- Aktifasi SSL

Sebelumnya kita buat dulu file `dhparam.pem` dengan perintah berikut:

```
openssl dhparam -C -2 -out /etc/ssl/certs/dhparam.pem 4096
```

Selanjutnya tambahkan konfig pada file `myapps.conf`

```
cd /etc/httpd/conf.d/
vim myapps.conf
```

tambahkan di bawahnya sebagai berikut:

```
<VirtualHost *:443 >
    ServerName "myapps.co.id"
    ServerAlias "www.myapps.co.id"
    ServerAlias "apps.myapps.co.id"
    UseCanonicalName Off

    SSLEngine on
    DocumentRoot "/home/myapps/httpdocs"
    CustomLog /home/myapps/logs/access_log combined
    ErrorLog /home/myapps/logs/error_log
    DirectoryIndex index.html index.php

    <IfModule mod_suexec.c>
        SuexecUserGroup "myapps" "myapps"
    </IfModule>

    <FilesMatch \.php$>
        SetHandler "proxy:unix:/run/php56-fpm.sock|fcgi://localhost/"
    </FilesMatch>

    Timeout 120
    <Directory /home/myapps/httpdocs>
```

```
#Options +Indexes +FollowSymLinks +MultiViews
AllowOverride All
Order allow,deny
allow from all
Require all granted

</Directory>

#<IfModule mod_rewrite.c>
# RewriteEngine On
# RewriteCond %{HTTP_HOST} ^myapps.co.id$ [NC]
# RewriteRule ^(.*)$ http://www.myapps.co.id$1 [L,R=301]
#</IfModule>

Include "/etc/httpd/include/myapps_ssl.conf"

</VirtualHost>
```

Nah, di situ ada juga file `myapps_ssl.conf` kita buat juga file-nya:

```
mkdir -p /etc/httpd/include
vim /etc/httpd/include/myapps_ssl.conf
```

Isinya kurang lebih seperti berikut:

```
SSLCertificateFile /etc/ssl/myapps/www-myapps-co-id.crt
SSLCertificateKeyFile /etc/ssl/myapps/www-myapps-co-id.key
SSLCertificateChainFile /etc/ssl/myapps/CA-fullchain.pem

SSLProtocol TLSv1.2
SSLCompression off
SSLProtocol All -SSLv2 -SSLv3
SSLHonorCipherOrder on
SSLCipherSuite "EECDH+ECDSA+AESGCM EECDH+aRSA+AESGCM EECDH+ECDSA+SHA384
EECDH+ECDSA+SHA256 EECDH+aRSA+SHA384 EECDH+aRSA+SHA256 EECDH EDH+aRSA
!CAMELLIA !SEED !3DES !RC4 !aNULL !eNULL !LOW !MD5 !EXP !PSK !SRP !DSS"
#SSLOpenSSLConfCmd Curves secp384r1
#SSLOpenSSLConfCmd DHParameters "/etc/ssl/certs/dhparam.pem"

<FilesMatch "\.(ttf|ttc|otf|eot|woff|woff2|font.css|css|js)$">
    Header always set Access-Control-Allow-Origin "*"
</FilesMatch>
```

Tentunya, berkas yang ada di `/etc/ssl/myapps/*` perlu dipersiapkan terlebih dahulu, yaitu file SSL yang telah disediakan dari klien.

- Konfigurasi httpd untuk security

Buat sebuah file `security.conf` secara terpisah:

```
cd /etc/httpd/include
vim security.conf
```

Isinya kurang lebih seperti berikut:

```
ServerTokens Prod
ServerSignature Off
TraceEnable Off

Header set X-Frame-Options: "SAMEORIGIN"
Header set X-Content-Type-Options: "nosniff"
Header set X-XSS-Protection: "1; mode=block"
Header always edit Set-Cookie (.*) "$1;HttpOnly;Secure"
Header always set Strict-Transport-Security "max-age=31536000;
includeSubDomains"
# sesuaikan bagian ini
Header always set Content-Security-Policy "default-src 'self'
https://*.piwik.pro www.google-analytics.com stats.g.doubleclick.net
dialogflow.cloud.google.com; font-src 'self' data:
https://fonts.gstatic.com; img-src 'self' data:
https://www.google-analytics.com https://s
tats.g.doubleclick.net *.myapps.co.id https://*.piwik.pro
https://*.google.com https://*.google.co.id *.googleusercontent.com; script-
src * https://www.googletagmanager.com www.google-analytics.com
https://*.piwik.pro 'unsafe-inline' 'unsafe-eval'; style-src * 'unsafe-inli
ne'; frame-src https://*.youtube.com https://app.midtrans.com
https://app.sandbox.midtrans.com https://bid.g.doubleclick.net *.qiscus.com;
connect-src 'self' *.myapps.co.id *.qiscus.com *.google-analytics.com
*.doubleclick.net dialogflow.cloud.google.com;"
Header always set Referrer-Policy "strict-origin"
Header always set Feature-Policy: "vibrate 'none'"
```

Lalu sertakan berkas tersebut ke dalam konfigurasi httpd yang utama, yaitu di httpd.conf:

```
vim /etc/httpd/conf/httpd.conf
```

Tambahkan teks berikut di akhir baris:

```
Include "/etc/httpd/include/security.conf"
```

Simpan, lalu cek untuk memastikan tidak ada konfigurasi yang error dengan perintah:

```
apachectl configtest
```

Jika muncul keluaran Syntax OK, maka bisa dilanjutkan untuk restart service httpd-nya.

```
systemctl restart httpd
systemctl status httpd
```

Sekian catatan kali ini.

From:
<https://wiki.samsul.web.id/> - **Samsul Maarif**

Permanent link:
<https://wiki.samsul.web.id/linux/Install.Laravel.dengan.php.5.6.di.CentOS.7>

Last update: **2022/12/29 04:41**

