Zu Testzwecken und in Produktven Umgebungen ist es oft nötig, mehrere Webseiten mit unterschiedlichen PHP-Versionen zu hosten.

Hier eine Anleitung, wie man dieses einrichtet.

Mehrere PHP - Versionen installieren

Zunächst müssen wir uns erst einmal ein Repositoriy besorten, das verschiedene (aktuelle) PHP-Version hat.

Es gibt das Remi-Repo (https://rpms.remirepo.net/) und das Software-Collection-Repo (https://www.softwarecollections.org/en/). Ich habe mich für das Remi-Repo entschieden, da es aktuellere Software beinhaltet (gerade bei PHP ist das von Vorteil) und es hier auch php7.3 gibt.

Remi-Repo einrichten

Ich benutze immer den Wizzard (https://rpms.remirepo.net/wizard/), um das Repo einzurichten. Vorteil: Richtige Url's und der richtige Syntax wird vorgegeben.

Paketinstallation

Für ein Installation von mehreren PHP-Versionen müssen wir nicht die Standard-PHP-Pakete (`yum list php-*`) installieren.

Je nach dem, welche PHP-Version man möchte, heißen die Pakete jetzt: php<Version>-php-<modul>

PHP 5.4

```
yum install php54-php php54-php-bcmath php54-php-cli php54-php-common php54-php-devel php54-php-gd  php54-php-gmp.x86_64 php54-php-intl.x86_64 php54-php-json.x86_64 php54-php-mbstring.x86_64 php54-php-mcrypt.x86_64 php54-php-mysqlnd.x86_64 php54-php-opcache.x86_64 php54-php-pdo.x86_64  php54-php-process.x86_64 php54-php-tidy.x86_64 php54-php-xml.x86_64 php54-php-pecl-apcu.x86_64 php54-php-pecl-geoip.x86_64 php54-php-pecl-igbinary.x86_64 php54-php-pecl-imagick.x86_64 php54-php-pecl-memcached.x86_64 php54-php-pecl-msgpack.x86_64 php54-php-pecl-zip.x86_64
```

PHP 5.5

```
yum install php55-php php55-php-bcmath php55-php-cli php55-php-common php55-php-devel php55-php-gd  php55-php-gmp.x86_64 php55-php-intl.x86_64 php55-php-json.x86_64 php55-php-mbstring.x86_64 php55-php-mcrypt.x86_64 php55-php-mysqlnd.x86_64 php55-php-opcache.x86_64 php55-php-pdo.x86_64  php55-php-
```

```
process.x86_64 php55-php-tidy.x86_64 php55-php-xml.x86_64 php55-php-pecl-
apcu.x86_64 php55-php-pecl-geoip.x86_64 php55-php-pecl-igbinary.x86_64
php55-php-pecl-imagick.x86_64 php55-php-pecl-memcached.x86_64 php55-php-
pecl-msgpack.x86_64 php55-php-pecl-zip.x86_64
```

PHP 5.6

```
yum install php56-php php56-php-bcmath php56-php-cli php56-php-common php56-
php-devel php56-php-gd  php56-php-gmp.x86_64 php56-php-intl.x86_64 php56-
php-json.x86_64 php56-php-mbstring.x86_64 php56-php-mcrypt.x86_64 php56-php-
mysqlnd.x86_64 php56-php-opcache.x86_64 php56-php-pdo.x86_64  php56-php-
process.x86_64 php56-php-tidy.x86_64 php56-php-xml.x86_64 php56-php-pecl-
apcu.x86_64 php56-php-pecl-geoip.x86_64 php56-php-pecl-igbinary.x86_64
php56-php-pecl-imagick.x86_64 php56-php-pecl-memcached.x86_64 php56-php-
pecl-msgpack.x86_64 php56-php-pecl-zip.x86_64
```

Notiz:

Ab hier wird nur noch auf die PHP-Versionen 7.x eingegangen. PHP in der Version 5.x sollte nicht mehr verwendet werden, wird der vollständigkeit halber aber aufgelistet

PHP 7.0

```
yum install php70-php php70-php-bcmath php70-php-cli php70-php-common php70-
php-devel php70-php-gd  php70-php-gmp.x86_64 php70-php-intl.x86_64 php70-
php-json.x86_64 php70-php-mbstring.x86_64 php70-php-mcrypt.x86_64 php70-php-
mysqlnd.x86_64 php70-php-opcache.x86_64 php70-php-pdo.x86_64  php70-php-
process.x86_64 php70-php-tidy.x86_64 php70-php-xml.x86_64 php70-php-pecl-
apcu.x86_64 php70-php-pecl-geoip.x86_64 php70-php-pecl-igbinary.x86_64
php70-php-pecl-imagick.x86_64 php70-php-pecl-memcached.x86_64 php70-php-
pecl-msgpack.x86_64 php70-php-pecl-zip.x86_64
```

PHP 7.1

```
yum install php71-php php71-php-bcmath php71-php-cli php71-php-common php71-
php-devel php71-php-gd  php71-php-gmp.x86_64 php71-php-intl.x86_64 php71-
php-json.x86_64 php71-php-mbstring.x86_64 php71-php-mcrypt.x86_64 php71-php-
mysqlnd.x86_64 php71-php-opcache.x86_64 php71-php-pdo.x86_64  php71-php-
process.x86_64 php71-php-tidy.x86_64 php71-php-xml.x86_64 php71-php-pecl-
apcu.x86_64 php71-php-pecl-geoip.x86_64 php71-php-pecl-igbinary.x86_64
php71-php-pecl-imagick.x86_64 php71-php-pecl-memcached.x86_64 php71-php-
pecl-msgpack.x86_64 php71-php-pecl-zip.x86_64
```

PHP 7.2

```
yum install php72-php php72-php-bcmath php72-php-cli php72-php-common php72-php-devel php72-php-gd  php72-php-gmp.x86_64 php72-php-intl.x86_64 php72-php-json.x86_64 php72-php-mbstring.x86_64 php72-php-mcrypt.x86_64 php72-php-mysqlnd.x86_64 php72-php-opcache.x86_64 php72-php-pdo.x86_64  php72-php-process.x86_64 php72-php-tidy.x86_64 php72-php-xml.x86_64 php72-php-pecl-apcu.x86_64 php72-php-pecl-geoip.x86_64 php72-php-pecl-igbinary.x86_64 php72-php-pecl-imagick.x86_64 php72-php-pecl-memcached.x86_64 php72-php-pecl-msgpack.x86_64 php72-php-pecl-zip.x86_64
```

PHP 7.3

```
yum install php73-php php73-php-bcmath php73-php-cli php73-php-common php73-php-devel php73-php-gd  php73-php-gmp.x86_64 php73-php-intl.x86_64 php73-php-json.x86_64 php73-php-mbstring.x86_64 php73-php-mcrypt.x86_64 php73-php-mysqlnd.x86_64 php73-php-opcache.x86_64 php73-php-pdo.x86_64  php73-php-process.x86_64 php73-php-tidy.x86_64 php73-php-xml.x86_64 php73-php-pecl-apcu.x86_64 php73-php-pecl-geoip.x86_64 php73-php-pecl-igbinary.x86_64 php73-php-pecl-imagick.x86_64 php73-php-pecl-memcached.x86_64 php73-php-pecl-msgpack.x86_64 php73-php-pecl-zip.x86_64
```

PHP 7.4

```
yum install $(yum list installed "php??-*" | awk '{print $1}' | egrep -v "Installed|Loaded" | cut -d "-" -f 2- | cut -d "." -f 1 | sort | uniq | egrep -v "php7|runtime" | sed -e 's/^/php74-/')
```

PHP FastCGI Process Manager

Jetzt kommt der Unterschied zur normalen Installation.

Wir installieren den PHP FastCGI Process Manager für jede PHP-Version. Dieser wird später vom Webserver aufgerufen wenn es dynamische Inhalte zu verarbeiten gibt. Der Process Manager verarbeitet diese und gibt alles wieder an den Webserver zurück.

Paketinstallation

```
yum install  php70-php-fpm.x86_64 php71-php-fpm.x86_64 php72-php-fpm.x86_64 php73-php-fpm.x86_64
```

Prozesse stoppen

```
systemctl stop php70-php-fpm.service php71-php-fpm.service php72-php-fpm.service php73-php-fpm.service
```

Wenn man mehrere FastCGI Process Manager installiert hat, kann es in de Prozessansicht ziemlich unübersichtlich werden, da jeder mit einem Pool namens www arbeitet. Deshalb werden die Pool's umbenannt.

Auch benutzt jeder Process Manager den gleichen Port. Dieser wird auch umgelegt

PHP-Version	Pool-Name	Port
7.0	www-php70	60070
7.1	www-php71	60071
7.2	www-php72	60072
7.3	www-php73	60073

Pool-Namen umbenennen

```
sed -e 's/\[www\]/\[www-php7.0\]/' -i /etc/opt/remi/php70/php-fpm.d/www.conf
sed -e 's/\[www\]/\[www-php7.1\]/' -i /etc/opt/remi/php71/php-fpm.d/www.conf
sed -e 's/\[www\]/\[www-php7.2\]/' -i /etc/opt/remi/php72/php-fpm.d/www.conf
sed -e 's/\[www\]/\[www-php7.3\]/' -i /etc/opt/remi/php73/php-fpm.d/www.conf
```

Port's umbenennen

```
sed -e 's/\:9000/\:60070/' -i /etc/opt/remi/php70/php-fpm.d/www.conf
sed -e 's/\:9000/\:60071/' -i /etc/opt/remi/php71/php-fpm.d/www.conf
sed -e 's/\:9000/\:60072/' -i /etc/opt/remi/php72/php-fpm.d/www.conf
sed -e 's/\:9000/\:60073/' -i /etc/opt/remi/php73/php-fpm.d/www.conf
```

Zeitzonen einer Jeden PHP-Version umbennenen

```
sed -e 's/;date.timezone =/;date.timezone =\ndate.timezone =
Europe\/Berlin/' -i /etc/opt/remi/php70/php.ini
sed -e 's/;date.timezone =/;date.timezone =\ndate.timezone =
Europe\/Berlin/' -i /etc/opt/remi/php71/php.ini
sed -e 's/;date.timezone =/;date.timezone =\ndate.timezone =
Europe\/Berlin/' -i /etc/opt/remi/php72/php.ini
sed -e 's/;date.timezone =/;date.timezone =\ndate.timezone =
Europe\/Berlin/' -i /etc/opt/remi/php73/php.ini
```

Startverhalten ändern

Standardmäßig wird jeder pool 5 mal gestartet. Ist das notwendig?

Es gibt die Option, den pool „on demand“ zu starten

Nachtrag:

Nach mehreren Test war ondemand 15% - 25% schneller als dynamisch

```
sed -e 's/pm = dynamic/;pm = dynamic\npm = ondemand/' -i
/etc/opt/remi/php70/php-fpm.d/www.conf
sed -e 's/pm = dynamic/;pm = dynamic\npm = ondemand/' -i
/etc/opt/remi/php71/php-fpm.d/www.conf
sed -e 's/pm = dynamic/;pm = dynamic\npm = ondemand/' -i
/etc/opt/remi/php72/php-fpm.d/www.conf
sed -e 's/pm = dynamic/;pm = dynamic\npm = ondemand/' -i
/etc/opt/remi/php73/php-fpm.d/www.conf
```

SELinux

Zum schluss solltem man noch SELinux sagen, dass es sich bei den Port's, die der Processmanager benutzt, um autorisierte http-Ports dreht.

```
semanage port -a -t http_port_t -p tcp 60070
semanage port -a -t http_port_t -p tcp 60071
semanage port -a -t http_port_t -p tcp 60072
semanage port -a -t http_port_t -p tcp 60073
```

FastCGI Process Manager Starten und enable'n

Fast fertig

Jetzt noch alle Prozessmanager starten und beim boot aktivieren

```
systemctl start php70-php-fpm.service php71-php-fpm.service php72-php-
fpm.service php73-php-fpm.service
systemctl enable php70-php-fpm.service php71-php-fpm.service php72-php-
fpm.service php73-php-fpm.service
```

Benutzung

Ab sofort kann man den FastCGI Process Manager benutzen. Wie?

Indem man jetzt in seiner httpd-Config bei den „Directory“-Blöcken den Process Manager aufruft

Beispiel

```
        #PHP in Version 7.0
        <Directory /var/www/html/phptest/php-7.0/>
                <FilesMatch \.php$>
```

```
                SetHandler "proxy:fcgi://127.0.0.1:60070"
        </FilesMatch>
</Directory>

#PHP in Version 7.1
<Directory /var/www/html/phptest/php-7.1/>
        <FilesMatch \.php$>
            SetHandler "proxy:fcgi://127.0.0.1:60071"
        </FilesMatch>
</Directory>
```

Nachteile

Problem: php kann auf der Konsole nicht mehr aufgerufen werden

Lösung: Alle php-Versionen sind verfügbar und können über php70, php71, php72, php73 aufgerufen werden. Es handelt sich hier um symbolische Links

```
ls -la /bin/php70 /bin/php71 /bin/php72 /bin/php73
lrwxrwxrwx 1 root root      32 Jan 30 14:40 /bin/php70 ->
/opt/remi/php70/root/usr/bin/php
lrwxrwxrwx 1 root root      32 Jan 30 14:35 /bin/php71 ->
/opt/remi/php71/root/usr/bin/php
lrwxrwxrwx 1 root root      32 Jan 30 14:36 /bin/php72 ->
/opt/remi/php72/root/usr/bin/php
lrwxrwxrwx 1 root root      32 Jan 30 14:41 /bin/php73 ->
/opt/remi/php73/root/usr/bin/php
```

Um nicht alle Scripte umschreiben zu müssen, kann man php über einen symbolischen Link erzeugen: `ln -s /opt/remi/php72/root/usr/bin/php /bin/php`

PHP deinstallieren

Sollten noch irgendwo Standard-PHP Paket installiert sein, findet man sie über die folgende Komando-Zeile

```
yum list installed php-* | cut -d " " -f 1 | grep ^php- | tr -s '\n' ' '
```

All diese Pakete kann man bedenkenlos deinstallieren

Quellen

- PHP Configuration Tips

- Using PHP 7.x on CentOS 7.4
- Apache with various PHP versions, using SCL
- https://blog.remirepo.net/post/2016/04/16/My-PHP-Workstation
- Website Speed test

From:
https://wiki.da-checka.de/ - **PSwiki**

Permanent link:
https://wiki.da-checka.de/doku.php/wiki/centos/mehrere_php_versionen?rev=1586943142

Last update: **2020/04/15 11:32**