PHP

- Composer Fatal error: Allowed memory size of XXXXX
- Cambiar la versión PHP para el shell (composer)
- Testing con operaciones de número flotante: eventos astronómicos

Composer Fatal error: Allowed memory size of XXXXX

En servidores cPanel con CloudLinux puede haber problemas para usar uno de los sistemas recomendados por Composer puede no funcionarnos y comeinza la frustación.

Ejecución normal

```
$  composer update
Loading composer repositories with package information
Warning from https://repo.packagist.org: You are using an outdated version of Composer.
Composer 2 is now available and you should upgrade. See https://getcomposer.org/2
Updating dependencies (including require-dev)

Fatal error: Allowed memory size of 1610612736 bytes exhausted (tried to allocate 528384
bytes) in phar:///usr/local/bin/composer1/src/Composer/DependencyResolver/Pool.php on line 230

Check https://getcomposer.org/doc/articles/troubleshooting.md#memory-limit-errors for more
info on how to handle out of memory errors.Fri May  7 12:24:49 CEST 2021
```

Ejecución con variable de entorno

```
$ COMPOSER_MEMORY_LIMIT=-1 composer <arguments>
```

Ejecución con argumentos

En algunos escenarios puede fallarnos, por ejemplo cuando tenemos dos instancias de composer (composer v1 y composer v2) y además el sistema tiene Cloudlinux. Podemos usar la siguiente alternativa

```
$ php -d memory_limit=-1 composer <arguments>
```

Información original

Como en todo la documentación original es importante conocerla.

- Composer Documentation - Memory limit errors

Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como esta, sin que ello implique ningún obligación ni responsabilidad por parte de Castris

Si necesitas soporte profesional puedes contratar con Castris soporte profesional.

Cambiar la versión PHP para el shell (composer)

Introducción

Cuando un sistema linux tiene múltiples versiones de PHP (**cPanel**, Directadmin, sistema sin panel de control, etc) uno de los problemas habituales es como usar una versión php en la shell diferente de la que está por defecto en el sistema. Bien, puedes especificar una versión por usuario e incluso puedes hacer que el usuario use distintas versiones PHP en el mismo usuario para distintos proyectos y sin Docker, como al trabvajar con Laravel, Symfony y otros.

Archivo de configuración de mi shell

El ejemplo en este caso es para bash, pero poco difiere de otros shell (yo en lo particular uso **zsh**) ya que todos tienen en común un fichero de configuración que se carga cuando iniciamos sesión o cuando tras modificarlo, ejecutamos un comando para que vuelva a leer ese fichero.

- .bash_profile
- .bashrc
- .kshrc
- .zshrc
- .tcshrc

Cambiar la versión PHP de tu usuario de shell en bash

Iniciar sesión con tu usuario. Asegurarnos de que estamos en el inicio de sesión

```
$ cd ~
```

Obtener el path de las versiones instaladas

```
# ls -lisah /usr/local/bin/php
19270949 0 lrwxrwxrwx 1 root root 26 Jun  8 08:39 /usr/local/bin/php ->
```

```
/usr/local/php56/bin/php56
# ls -lisah /usr/local
ls /usr/local/
...  php71  php73  php80 ... php56   php72  php74
```

Agregar la línea para usar la versión requerida

```
# .bashrc o .bash_profile Véase su distribución
# .zshrc
export PATH=/usr/local/php80/bin:$PATH
```

Cpanel (v98+)

En cpanel las instalaciones de php estan en el directorio `/opt/cpanel/` con la nomemclatura `ea-phpXX` para indicar la version por lo que el path a añadir sería `export PATH=/opt/cpanel/ea-php80/root/bin/:$PATH` por ejemplo

- ea-php74
- ea-php80

Salvar y hacer un reload de tu perfil

```
# bash
$ . ~/.bash_profile
# zsh
$ source ~/.zshrc
```

Comprobar la versión PHP de nuestro shell

```
php -v
PHP 8.0.5 (cli) (built: May  3 2021 11:30:57) ( NTS )
Copyright (c) The PHP Group
Zend Engine v4.0.5, Copyright (c) Zend Technologies
 with Zend OPcache v8.0.5, Copyright (c), by Zend Technologies
```

Usar alias para distintos proyectos con el mismo usuarios

En mi caso como dije uso zsh y tengo múltiples proyectos. No me gusta marearme con docker salvo para cosas específicas asi que prefiero tener unos alias en mi fichero `~/.zshrc`

```
alias p74='/usr/bin/php7.4 '
alias a74='/usr/bin/php7.4 artisan '
alias c74='/usr/bin/php7.4 /usr/local/bin/composer '
```

Ejemplo usar versión especifica de PHP en el shell de una cuenta cpanel

Editar el fochero .bashrc del usuario

```
...
# Uncomment the following line if you don't like systemctl's auto-paging feature:
# export SYSTEMD_PAGER=
...
alias php=/opt/cpanel/ea-php80/root/usr/bin/php
alias composer='/opt/cpanel/ea-php80/root/usr/bin/php /opt/cpanel/composer/bin/composer'
```

Aviso

Testing con operaciones de número flotante: eventos astronómicos

Introducción

En computación, los cálculos numéricos, especialmente los que implican flotantes, pueden ser propensos a pequeñas imprecisiones. Esto se debe a la forma en que las computadoras manejan la aritmética de punto flotante. Las imprecisiones pueden acumularse en cálculos complejos, lo que puede llevar a resultados ligeramente fuera de lo esperado.

En el caso de una aplicación que estaba haciendo de repente unos test me comenzaron a fallar justo cuando habia realizado unos cambios. Pero menos mal que me fije en los datos del fallo del test.

Era un simple segundo en la aserción de un datao astronomico calculado con el método de PHP `date_sun_info`

```
   FAILED  Tests\Unit\Observers\CommandCenterObserverTest > create command center fire
observer and save astro sun info
  Failed asserting that a row in the table [command_centers] matches the attributes {
    "sunrise": "2022-11-30 07:14:32",
    "sunset": "2022-11-30 16:46:11"
}.

Found similar results: [
    {
        "sunrise": "2022-11-30 07:14:32",
        "sunset": "2022-11-30 16:46:10"  << Fallo de un segundo
    }
].
```

Solución

En este tipo de casos en el que trabajamos con cuestiones que no requieren una exactitud matemática al más puro estilo de C++, podemos modificar nuestro test.

Parte del test original

```
$this->assertDatabaseHas('command_centers', [
    'sunrise' => '2022-11-30 07:14:32',
    'sunset' => '2022-11-30 16:46:11',
]);
```

Parte modificada del test

```
$expectedSunrise = Carbon::createFromFormat('Y-m-d H:i:s', '2022-11-30 07:14:32');
$actualSunrise = Carbon::createFromFormat('Y-m-d H:i:s', $commandCenter->sunrise->format('Y-m-d H:i:s'));

$this->assertLessThanOrEqual(
    1, // seconds of allowable difference
    $expectedSunrise->diffInSeconds($actualSunrise),
    'Sunrise time difference is more than 1 second'
);

$expectedSunset = Carbon::createFromFormat('Y-m-d H:i:s', '2022-11-30 16:46:11');
$actualSunset = Carbon::createFromFormat('Y-m-d H:i:s', $commandCenter->sunset->format('Y-m-d H:i:s'));

$this->assertLessThanOrEqual(
    1, // seconds of allowable difference
    $expectedSunset->diffInSeconds($actualSunset),
    'Sunset time difference is more than 1 second'
);
```

Aviso