

# Activar PHP8.2 JIT Compiler - Just-In-Time compilation (JIT)eva

## PHP8.2 JIT Compiler

Los lenguajes de programación puramente interpretados no tienen una fase de compilación y ejecutan directamente el código en una máquina virtual. La mayoría de los lenguajes interpretados, incluyendo PHP, de hecho, tienen una ligera fase de compilación para mejorar su rendimiento.

Por otro lado, los lenguajes de programación con compilación anticipada (AOT) requieren que el código sea compilado antes de su ejecución.

La compilación **Just-In-Time (JIT)** es un modelo híbrido entre el intérprete y la compilación anticipada, donde parte o todo el código se compila, a menudo en tiempo de ejecución, sin que el desarrollador tenga que compilarlo manualmente.

Históricamente, **PHP** fue un lenguaje interpretado, donde todo el código era interpretado por una máquina virtual (**Zend VM**). Esto cambió con la introducción de Opcache y los **Opcodes**, que se generaban a partir del código PHP y podían ser almacenados en memoria caché. PHP 7.0 introdujo el concepto de **AST (Árbol de Sintaxis Abstracta)**, que separó aún más el analizador sintáctico del compilador.

El JIT de PHP utiliza internamente **DynASM de LuaJIT** y está implementado como parte del **Opcache**.

El uso de JIT en PHP es altamente recomendable, no solo para propósito general, sino más aun, es aconsejable producir código de acuerdo a los cambios implementado desde PHP 8.0 que van mucho más allá de la activación de una forma de procesar en el servidor.

En particular lo uso en mis servidores en los que hago despliegues con Laravel y francamente, es impresionante.

# Activarlo en el servidor

En un servidor con PHP, en múltiples versiones, y con el modo FPM entre otros como uso con el servidor Web, y como ejemplo una Ubuntu, debemos activar JIT que por defecto no esta activado.

Encontraremos sus variables disponibles en el fichero `php.ini` de la versión en la que queremos activar JIT, pero como consejo, es mejor no hacerlo ahí.

En su lugar debemos hacerlo en `/etc/php/8.2/mods-available/` ya que es ahí donde están los ficheros parciales de cada modulo, y su configuración, tanto para PHP-FPM, como para **PHP en modo cli**.

“ Esta es una buena práctica que existe en Ubuntu y otras distribuciones de Linux.

El fichero en cuestión es `/etc/php/8.2/mods-available/opcache.ini` y contiene por defecto. Esto no activará JIT sino que lo cargará pero lo mostrará **apagado**, por ejemplo usando `phpinfo()`

```
; configuration for php opcache module
; priority=10
zend_extension=opcache.so
```

“ Si cambias los valores en el `php.ini` general es probable que todo este activado menos el par `JIT=On`

JIT Disabled  
Image not found or type unknown

## Configuración ejemplo

Bien, os dejo una configuración bastante funcional en un sistema con uso muy intensivo tanto de php vía colas de Laravel usando php-cli y vía API externa en producción.

“ Las mejoras son de 8,2 segundos en operaciones intensas a solo 1,2s.

```
zend_extension=opcache.so
opcache.jit=on
opcache.interned_strings_buffer=8
opcache.max_accelerated_files=10000
```

```
opcache.max_wasted_percentage=5
opcache.validate_timestamps=1
opcache.revalidate_freq=2
opcache.jit_buffer_size=100M
opcache.jit=tracing
opcache.jit=1255
opcache.save_comments=0
opcache.file_cache=/var/cache/php/opcache
```

Tras el reinicio de PHP-FPM verás algo parecido a esto:

**JIT Activado**  
Image could not be loaded or type unknown

“ Esta configuración `opcache.file_cache` requiere que exista `/var/cache/php/opcache` para servir de segundo nivel de cache (o el que tu definas), y que tenga los permisos para el ejecutor. En algunos sitios veo de usar `www-data` que es el que usa en una Ubuntu con Nginx. Puedes dejarlo vacío, si eres generoso con la memoria asignada. En realidad, es algo así como el swap en linux.

Te aconsejo una vuelta por el fichero `php.ini` principal, para que veas todos los comentarios de cada uno de los valores. Ya sabes, el copia y pega, puede ser fatal, pues lo que a mi me funciona puede que a ti no.

## Enlaces

- [Make your app faster with PHP 8.3](#)

## 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 se entrega, tal y como está, sin que ello implique ninguna obligación ni responsabilidad por parte de [Castris](#)

Si necesitas soporte profesional puedes contratar con Castris [soporte profesional](#).

---

Revision #1

Created 17 September 2024 15:49:50 by Abkrim

Updated 17 September 2024 16:18:44 by Abkrim