Let's Migrate to PHP 7

Slavey Karadzhov
Senior Consultant, Professional Services

Maurice Kherlakian
Senior Director, Professional Services

Presenter

Slavey Karadzhov

Senior Consultant

Rogue Wave Software

Slavey.Karadzhov@roguewave.com

© 2017 Rogue Wave Software, Inc. All Rights Reserved.

Presenter

- Zend Framework (ZF), PHP 5 and PHP 7 ZCE
- Trainer, consultant and helping hand
- Many years of PHP (v3 .. v7), ZF and galaxy of programming languages (Python, Perl, Java, JavaScript, C/C++ ...).
- Author of the “Learn ZF2” book: http://learnzf2.com

© 2017 Rogue Wave Software, Inc. All Rights Reserved.

Agenda

Agenda

- This workshop is about migrating to PHP 7
  - But a lot applies to migrations in general
- Migrations almost always catch us by surprise
  - What's the best strategy to approach a migration?
  - How to avoid the situation in the future
- Some of PHP 7's backward-incompatible changes
- How to monitor what's in production effectively

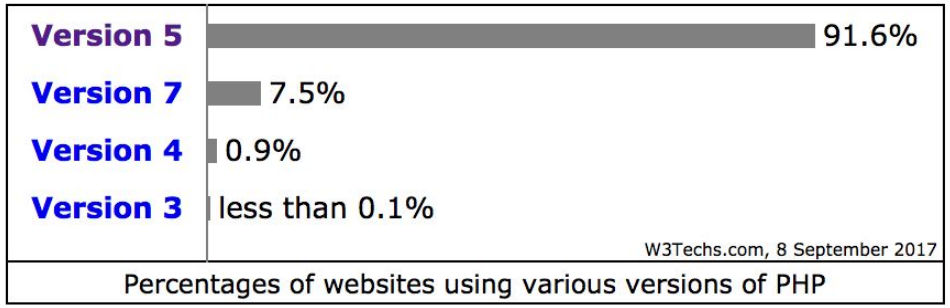
© 2017 Rogue Wave Software, Inc. All Rights Reserved.

Poll

Select the version of PHP that you're running in production.

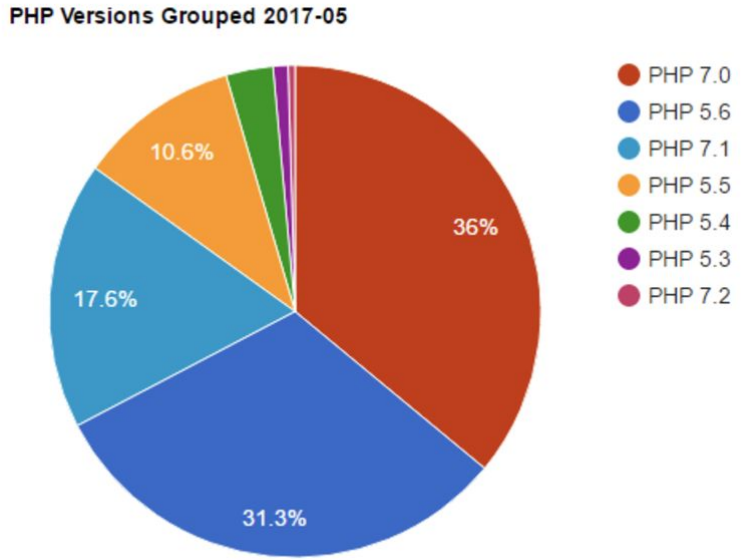
- PHP 7
- PHP 5.6
- PHP 5.5
- PHP 5.4
- Other version

How many are running PHP 7?



According to W3techs*

According to Composer**



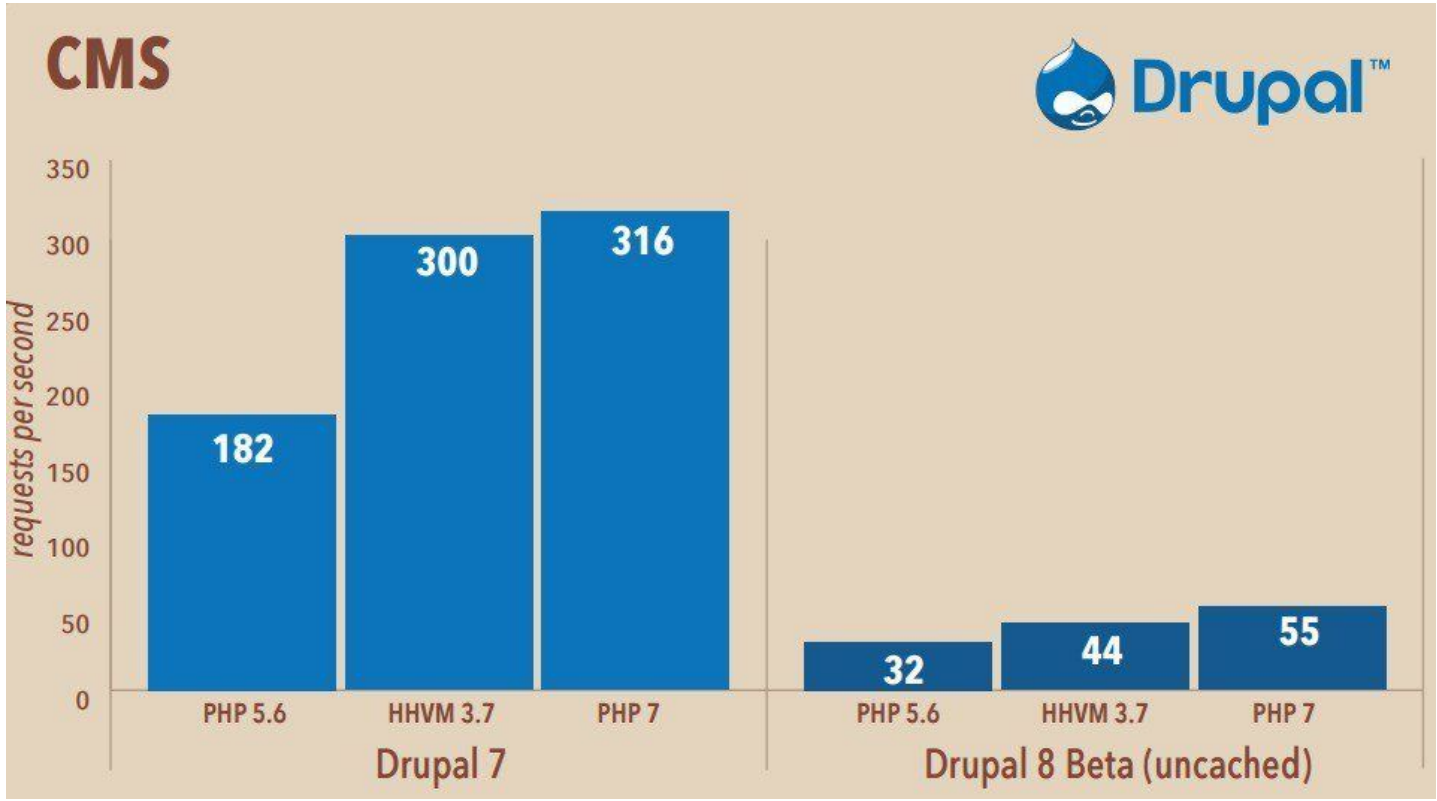
* https://w3techs.com/technologies/details/pl-php/all/all
** https://seld.be/notes/php-versions-stats-2017-1-edition

© 2017 Rogue Wave Software, Inc. All Rights Reserved.

Why Migrate to PHP 7?

Why Migrate to PHP 7?

- PHP 7 is fast!
  - Not only on paper but also in your real-life applications



© 2017 Rogue Wave Software, Inc. All Rights Reserved.

Why Migrate to PHP 7 (cont.d) ?

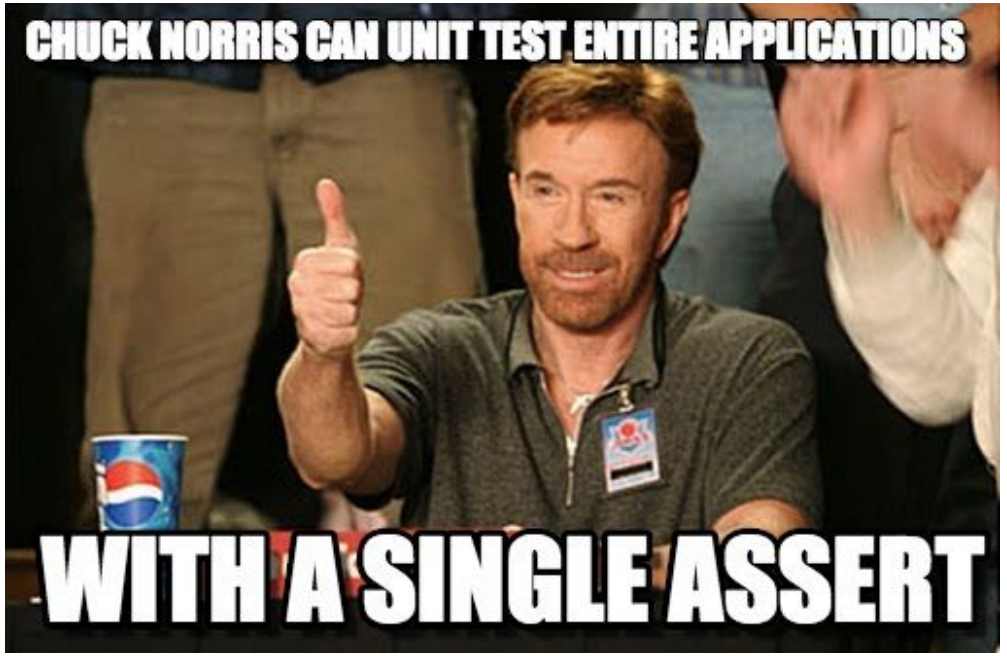
- No change in the current architecture or way of development
- It has stricter syntax and newer error handling

© 2017 Rogue Wave Software, Inc. All Rights Reserved.

What makes migration easier?

What makes migration easier?

1. A good plan
2. The right strategy for your situation
3. The right tools
4. TESTS!



© 2017 Rogue Wave Software, Inc. All Rights Reserved.

Migration strategies

Cut-over
(big bang)

Strangler
(StranglerApplication*)

* Martin Fowler - https://www.martinfowler.com/bliki/StranglerApplication.html

© 2017 Rogue Wave Software, Inc. All Rights Reserved.

How do you choose?

- Is your application fairly large? (100,000's LOC)
- Is your application mission critical?
- Can you afford bugs?

© 2017 Rogue Wave Software, Inc. All Rights Reserved.

Suggested way to migrate to PHP 7

- Keeping compatibility with PHP 5
- Make sure that the changed PHP source code works using PHP Lint

© 2017 Rogue Wave Software, Inc. All Rights Reserved.

Tools

- Version Control System (VCS)
  - git is preferable
- Syntax Checkers
- Static Code Analysis
  - help us discover and fix PHP7 migration issues
- Continuous Integration Systems
  - help us prevent regressions (regressions)

© 2017 Rogue Wave Software, Inc. All Rights Reserved.

Task 1: Put source code in VCS

Task 1: Put source code in VCS

- (If not present) create new repository for the PHP project
- (If not present) put the latest working code under the **master** branch
- Create a feature branch “feature/php7-migration”
- Start your work inside of the new “feature/php7-migration” branch
- (Optional) Use systems like GitLab to help you with VCS management

© 2017 Rogue Wave Software, Inc. All Rights Reserved.

Task 2: Activate syntax checker hook

Task 2: Activate syntax checker hook

- As git pre-commit hook
  - that is called for new or modified PHP files
- That uses php linter (php -l)

© 2017 Rogue Wave Software, Inc. All Rights Reserved.

Backwards-incompatible Changes

What changed? (backwards-incompatible changes)

- From 5.6, most changes are edge case, and subtle
- HOWEVER, they can be tough to find
- Complete list can be found here:

http://php.net/manual/en/migration70.incompatible.php

© 2017 Rogue Wave Software, Inc. All Rights Reserved.

Change: Indirect variable naming

Expression	PHP 5 interpretation	PHP 7 interpretation
$$foo['bar']['baz']	${$foo['bar']['baz']}	($$foo)['bar']['baz']
$foo->$bar['baz']	$foo->{$bar['baz']}	($foo->$bar)['baz']
$foo->$bar['baz']()	$foo->{$bar['baz']}()	($foo->$bar)['baz']()
Foo::$bar['baz']()	Foo::{$bar['baz']}()	(Foo::$bar)['baz']()

© 2017 Rogue Wave Software, Inc. All Rights Reserved.

Change: List handling

```php
<?php
list($a[], $a[], $a[]) = [1, 2, 3];
var_dump($a);
```

PHP 5.5

```
array(3) {
    [0]=>
    int(3)
    [1]=>
    int(2)
    [2]=>
    int(1)
}
```

PHP 7

```
array(3) {
    [0]=>
    int(1)
    [1]=>
    int(2)
    [2]=>
    int(3)
}
```


© 2017 Rogue Wave Software, Inc. All Rights Reserved.

Change: Variable variables no longer used with global

```php
<?php
function f() {
    // Valid in PHP 5 only.
    global $$foo->bar;

    // Valid in PHP 5 and 7.
    global ${$foo->bar};
}
?>
```


© 2017 Rogue Wave Software, Inc. All Rights Reserved.

Change: foreach doesn't change internal array pointer

```php
<?php

$arr = [1,2,3,4];
foreach($arr as $elt) {
    if($elt % 2 == 0) {
        break;
    }
}

$element = current($arr);

//php 5.5
2

//php 7
1
```


© 2017 Rogue Wave Software, Inc. All Rights Reserved.

Change: ext/mysql is gone

- No more mysql_*
- If you have a DBAL, replace the mysql_* calls
- If you don't, use short term fix:
  - Refactor all mysql_* calls to static methods, or a user defined function
  - Call mysqli_*
  - OR use one of the libraries:

  https://github.com/philip/MySQLConverterTool or

  https://github.com/dshafik/php7-mysql-shim (if you can install an extension on your server)
- Long term – refactor into proper DBAL with tests

© 2017 Rogue Wave Software, Inc. All Rights Reserved.

Changes are subtle

- Above changes are only PHP 5.6 => 7
- Code above will run but will result in bugs in very subtle ways
- In most cases, we migrate from 5.5, 5.4, or even 5.3
- Some tools can help make the job easier:
  - PHP CodeCniffer with Wim's PHPCompatibility https://github.com/wimg/PHPCompatibility

© 2017 Rogue Wave Software, Inc. All Rights Reserved.

Task 3: Install Static Code Analysis Tool

Task 3: Install Static Code Analysis Tool

- Install PHP Code Sniffer (v2.9)
  - pear install PHP_CodeSniffer-2.9.0
- Install PHPCompatibility Sniffs
  - https://github.com/wimg/PHPCompatibility
- Register the new sniffs
  - phpcs --config-set installed_paths /path/to/PHPCompatibility
- Start using it:
  - phpcs --standard=PHPCompatibility --runtime-set testVersion 7.1 -d memory_limit=-1 --ignore=*.js,*.css <path/to/php/application/>

© 2017 Rogue Wave Software, Inc. All Rights Reserved.

The Master Plan

The Master Plan

- Analyze
- Fix
- Test
- Merge
- Repeat

© 2017 Rogue Wave Software, Inc. All Rights Reserved.

The Master Plan (cont.d)

- If you can't afford any errors but you have no automated tests, there's another solution
- Remember the Strangler?
  - Migrate small pieces to the new PHP 7 syntax, testing with PHP 7 in development, but pushing to an older version of PHP
  - Use Code sniffer to detect issues in code
  - Repeat
  - You will end up with a PHP 7 codebase, but one for which you are pushing changes progressively and getting live user feedback.

© 2017 Rogue Wave Software, Inc. All Rights Reserved.

The Master Plan (cont.d)

- If this is still not enough, use feature toggles
  - Feature toggles are a great way to show new features to only subsets of users, or randomly selected users
  - Also a great way to A/B test
  - You can roll out your migrated code a few super users only, and then expand the pool when you are more confident

© 2017 Rogue Wave Software, Inc. All Rights Reserved.

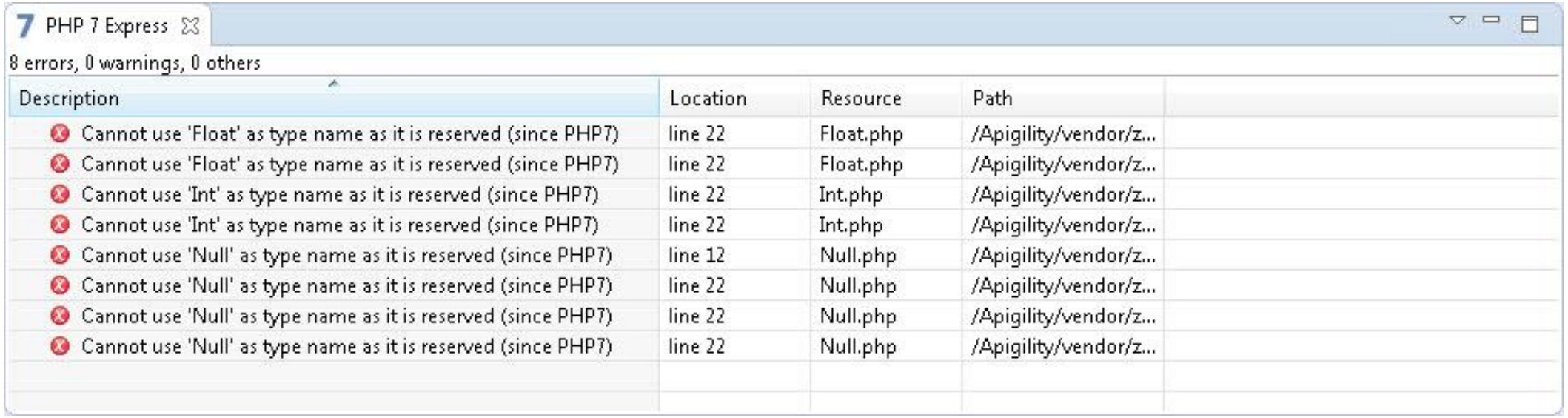
MP: Analyze

MP: Analyze

- Using PHP CodeSniffer
- Setup your IDE to recognise PHP 7 issues
- Using Zend Server Monitoring
- Using PHP error logs

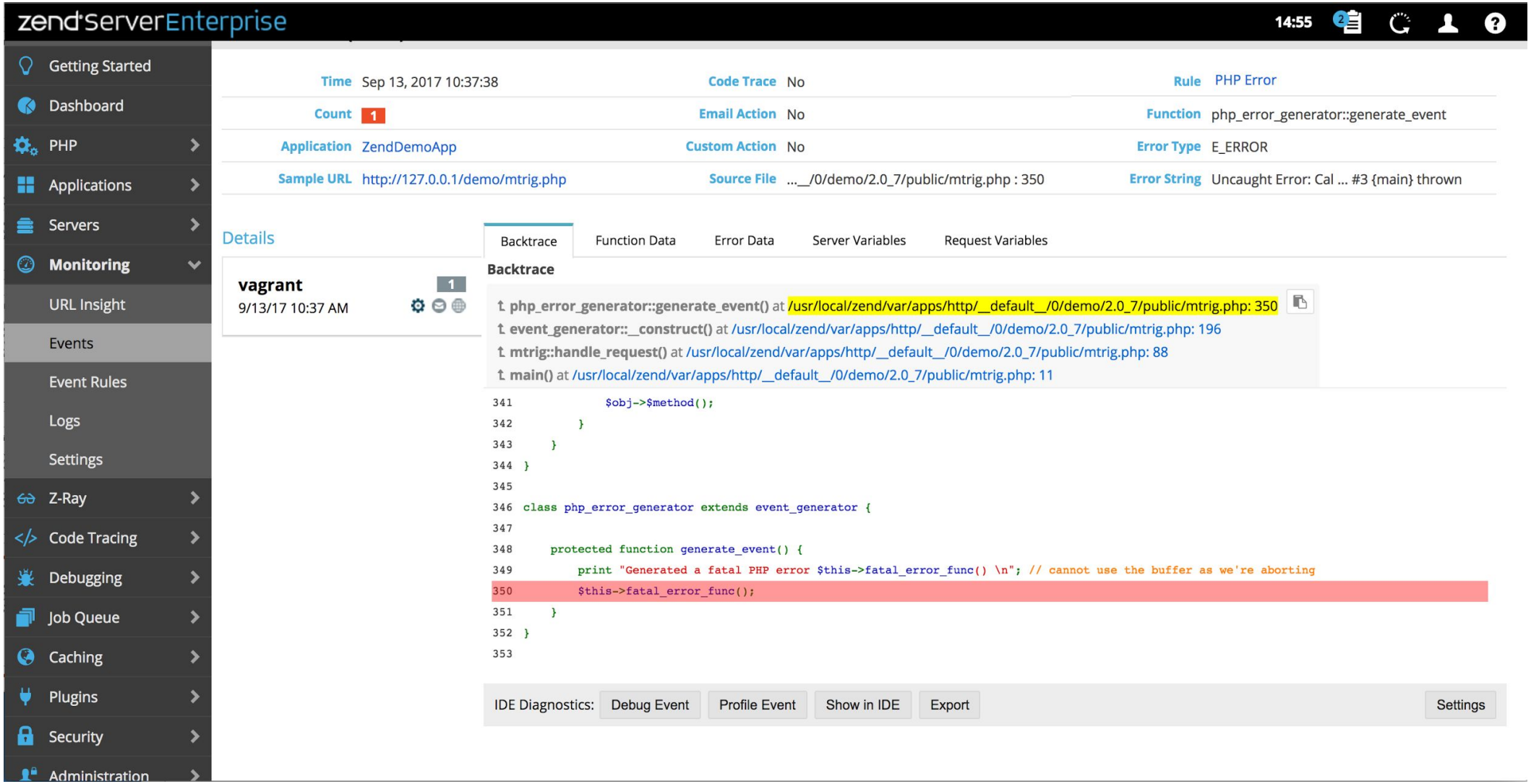
© 2017 Rogue Wave Software, Inc. All Rights Reserved.

MP: Analyze (cont.d)



© 2017 Rogue Wave Software, Inc. All Rights Reserved.

Monitoring



© 2017 Rogue Wave Software, Inc. All Rights Reserved.

Monitoring

- Ensure that your production systems are monitored
- This will help catch errors, especially if you phase your deployment to a subset of users
- Zend Server does exactly that – gives you contextual information about requests and shows you what went wrong

© 2017 Rogue Wave Software, Inc. All Rights Reserved.

MP: Fix

MP: Fix

- Own Source Code
  - Manually
  - Use PHP Code Sniffer for automated fixing of issues
- Libraries
  - (If possible) get the latest upstream versions
  - Enable composer and improve your architecture
    - Ex: composer require zendframework/zendframework1
  - and same options as own source code

© 2017 Rogue Wave Software, Inc. All Rights Reserved.

MP: Fix (cont.d)

- PHP Modules
  - Using "drop-in" replacements
  - Shims
  - Converters
  - Use PHP instead of C
  - ask Zend to help you ;)
  - or rewriting from scratch

- Examples:
  - mysql
    - Replacement: mysqli
    - Shim: https://github.com/dshafik/php7-mysql-shim

© 2017 Rogue Wave Software, Inc. All Rights Reserved.

MP: Fix (cont.d)

- Replacement strategies:
  - Long term – refactor into proper DBAL with tests
  - Short term:
    - Refactor all mysql_* calls to static methods, or a user defined function
    - Call mysqli_*
    - OR use one of the libraries:

https://github.com/philip/MySQLConverterTool or

https://github.com/dshafik/php7-mysql-shim

© 2017 Rogue Wave Software, Inc. All Rights Reserved.

Task 4: Automate “Clone” Fixes

Task 4: Automate “Clone” Fixes

- Create new Sniff only if it will save you time
- Sniffs require knowledge and patience

© 2017 Rogue Wave Software, Inc. All Rights Reserved.

MP: Fix (cont.d)

Automatically Fixable PHP 7 issues

- Parser errors now throw a ParseError object. Error handling for eval() should now include a catch block that can handle this error.
- clone($object)
- Class with PHP4 style constructor
- $this in static context
- Switch statements can not have multiple default blocks since PHP 7.0
- Assigning the return value of new by reference
- Call-time pass-by-reference
- split is no longer available (preg_split,explode, etc.)
- use php://input instead of HTTP_RAW_POST_DATA
- Long predefined variables have been removed (ex: '$HTTP_SERVER_VARS')
- preg_replace() - /e modifier is forbidden
- and more...

© 2017 Rogue Wave Software, Inc. All Rights Reserved.

MP: Fix (cont.d)

Difficult to fix

- Old “private” C module for PHP
- Object calls in static context ?!
- Non-standard "hacks"

```php
<?php

class X {

    public $parameter = "10";

    public function doX() {
        return Y::caller();
    }
}


class Y {
    public  function caller() {
        return $this->parameter;
    }
}


$x = new X();
echo $x->doX();
```


© 2017 Rogue Wave Software, Inc. All Rights Reserved.

MP: Test

MP: Test

- Test
  - PHP UnitTests, Functional tests, etc
  - PHPCodeSniffer
  - PHP Lint
- Automate
  - install Jenkins or cloud-based CI solutions.
  - automatic event notification from your monitoring system

© 2017 Rogue Wave Software, Inc. All Rights Reserved.

MP: Merge

MP: Merge

- Group things to merge by the issue(s) that they solve
- Merge tested and proven to work pieces back to the “master” branch
- Continue work on the “feature/php7-migration” branch.

And repeat: analyze, fix, test, merge.

© 2017 Rogue Wave Software, Inc. All Rights Reserved.

Q&A


RogueWave
SOFTWARE
Innovate with Confidence