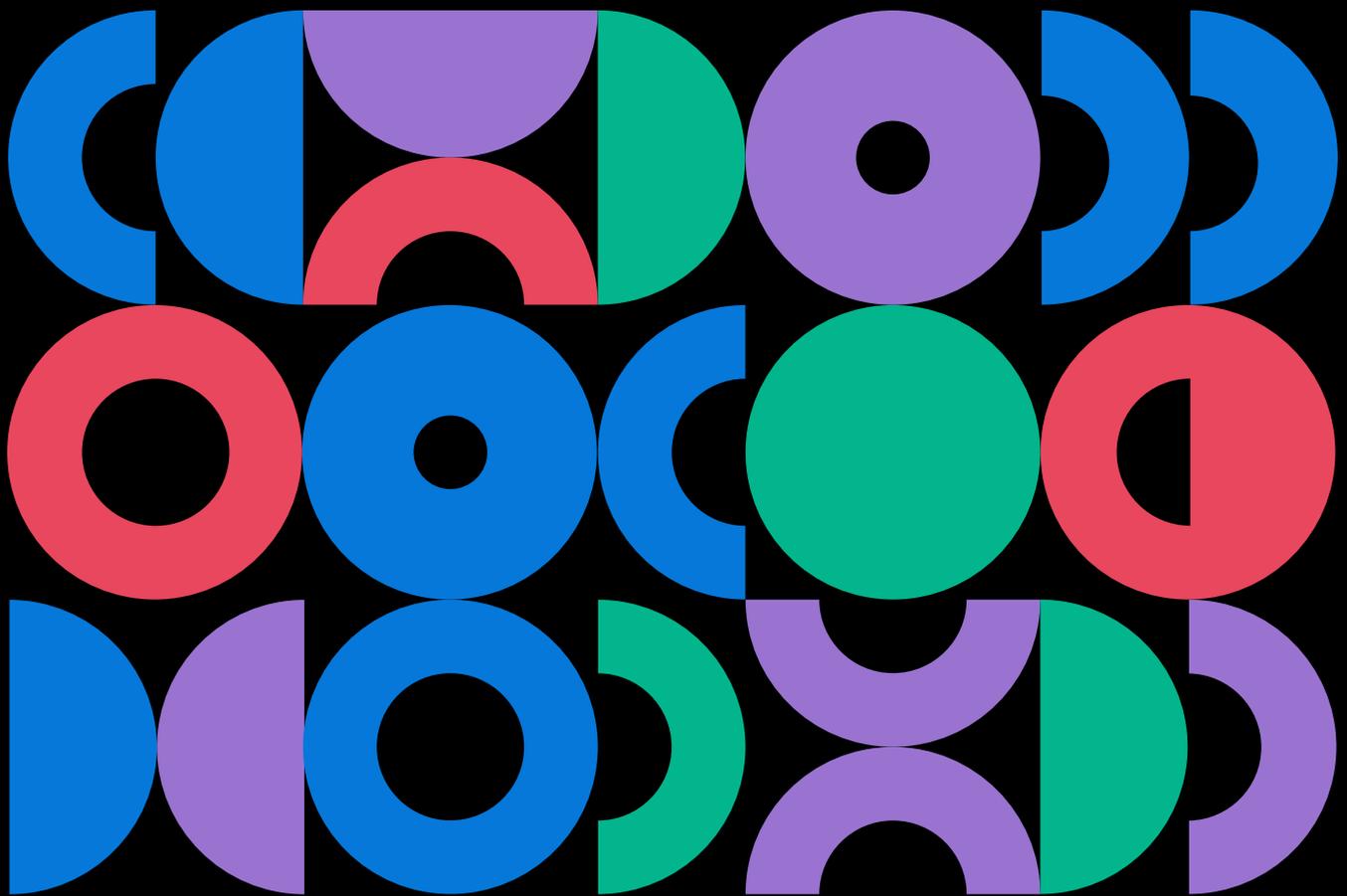




# 2023 PHP LANDSCAPE REPORT



# FOREWORD

## Welcome To The Era of PHP 8

To the reader,

November 2022 marked a milestone: with the end-of-life announcement of PHP 7.4, for the first time, PHP 8 is the only community supported PHP. While this was expected, and has been messaged for years with the publication of the [PHP lifecycle](#), it has huge impact.

As this survey shows, PHP 7.4 is still the most-used version of PHP, and many reported using even older versions. This leaves many users scrambling: do they upgrade their PHP version and take the time to make their application compatible, or do they look for long-term-support editions of the language in order to squeeze another year or two out of the PHP version they currently use? Migrating an application may seem like the best solution, but if an application has a finite lifespan, or a team is already at capacity, it's not always an option.

Another challenge facing PHP consumers comes in the form of infrastructure modernization. For the first time, a cloud provider (AWS) was the top choice of deployment platforms, edging out on-premise deployment. Additionally, three-quarters of respondents indicated they are now using containers. And relational databases, web APIs, and more are common integrations for production PHP applications. These shifts indicate that the days of FTP'ing files to a server in order to deploy a PHP application are long-gone. Organizations deploying PHP are looking for teams who can containerize applications and use sophisticated orchestration techniques in order to manage the entire infrastructure in which the application runs.

With these challenges, is PHP's popularity on the wane? To the contrary, our respondents rated PHP's ease of learning highly, as well as their overall enjoyment working with the language. The fact that they can integrate with a wide variety of technologies, get new features on a regular basis, and enjoy a steady pace of language performance improvements makes it a great fit for business critical applications and web APIs.

Welcome to the era of PHP 8!

Enjoy the report,  
Matthew Weier O'Phinney  
Product Manager  
Zend by Perforce

# TABLE OF CONTENTS

## 04

### About The Survey

Respondent Demographics .....	05
Development Team Size .....	06
Company Size .....	07

## 08

### Top PHP Application Types and Technologies

What Are Teams Developing With PHP .....	09
Top Application Integration Categories .....	10
Top Operating Systems For PHP Teams .....	11
Where Are Teams Deploying Their PHP Applications .....	13
Top Web Servers For PHP Teams .....	15
Compliance Requirements .....	16

## 17

### PHP Development and Maintenance

Top Development Priorities For PHP Teams .....	18
Maintenance vs. Feature Development .....	19
Finding Issues in Production .....	21
Solving Issues in Production .....	22

## 23

### Containerization and Orchestration

Container Usage .....	24
Top Container Technologies .....	26
Orchestration Usage .....	27
Top Orchestration Technologies .....	29

## 30

### PHP Version and Upgrade Statistics

Most Used PHP Versions .....	31
PHP Version Upgrade Planning .....	34
PHP Upgrade Challenges .....	35
Top Challenges For PHP Teams .....	36
Most Valuable PHP Features .....	37
The State of PHP .....	38

## 39

### Final Thoughts

About Zend .....	40
------------------	----

# ABOUT THE SURVEY

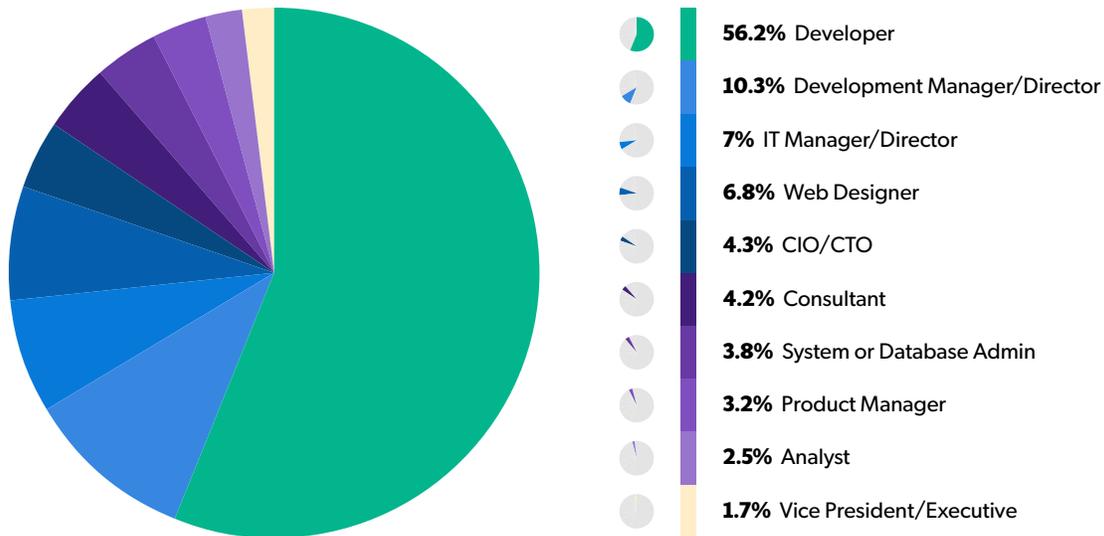
The Zend 2023 PHP Landscape Report is based on the results of an anonymous survey conducted between the months of October and December of 2022. The survey, which was promoted via social media and email, focused primarily on how PHP development teams are working with the language, their priorities in PHP development, versions they use and plans for upgrades, and the technologies they use in developing PHP applications.

After careful review of the responses, the survey received a total of 651 qualified responses.

# Respondent Demographics

Like previous years, the 2023 survey had a majority representation from developers. The top three respondent categories were developers, at 56.2%, development managers and directors, at 10.3%, and IT managers and directors, at 7%.

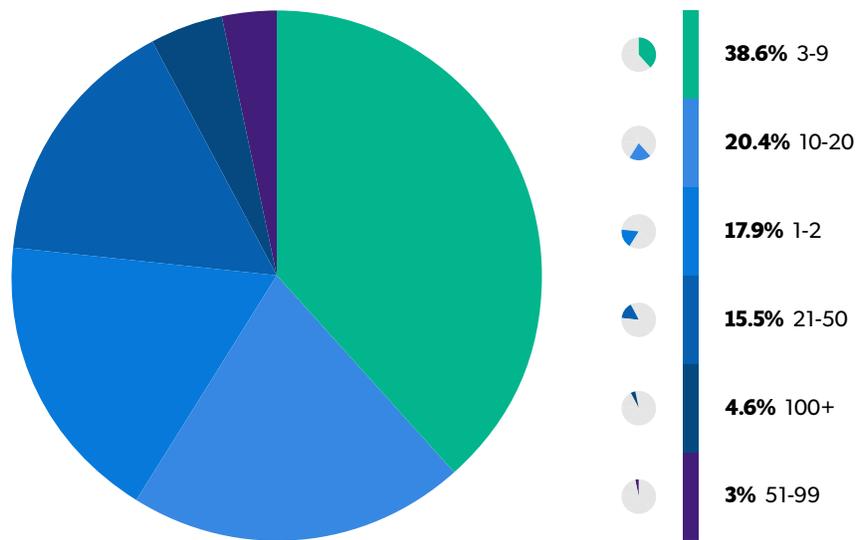
## Which Job Title Best Matches Your Current Role?



# Development Team Size

Continuing our demographic and firmographic questions, we asked respondents to share the size of their development teams. Like last year’s survey, the largest segment of respondents (38.6%) reported having a development team size of between 3-8 developers. The next highest segment, at 20.4%, was teams between 10-20 developers – marking significant representation for large or development-focused organizations. Overall, teams with more than 10 developers represented over 43% of all respondents.

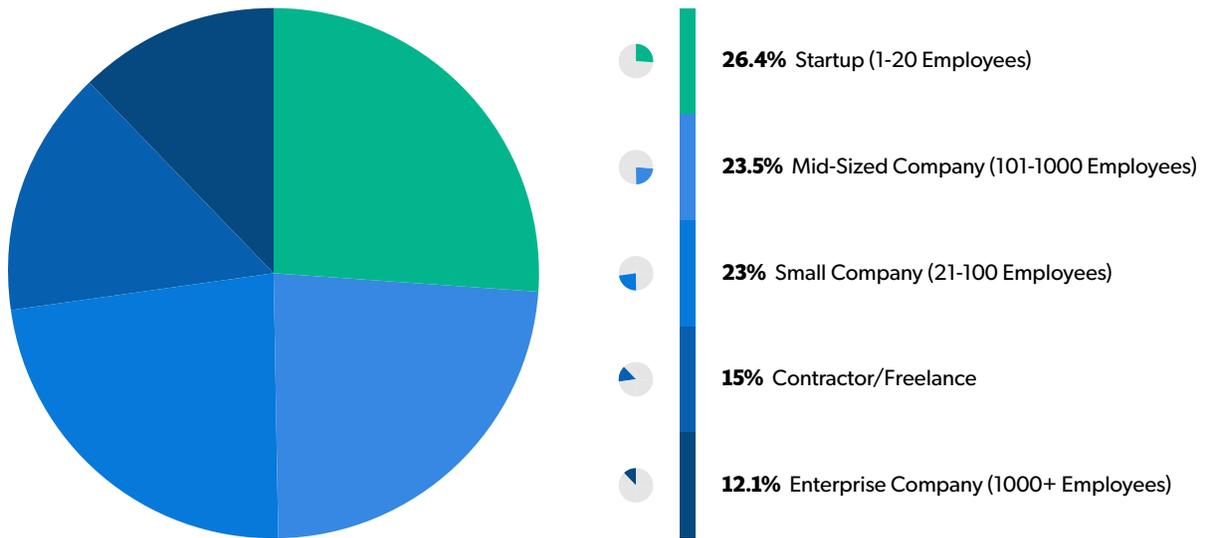
## What Is the Size of Your Development Team?



# Company Size

In our final firmographic/demographic question, we asked respondents to share the size of their company. As with our 2022 survey, we found a nearly even split between companies with 10-20, 21-100, and 101-1000 employees. Contractor/Freelance and Enterprise companies rounded out the results with 15% and 12.1% of respondents, respectively.

## What Is the Size of Your Company?



# TOP PHP APPLICATION TYPES & TECHNOLOGIES

While many know of PHP due to WordPress and other Content Management Systems (CMS), PHP has also long been a “glue” language for web applications and APIs. Developers frequently interact with other web APIs, relational databases, key-value stores, message queues, and more in order to deliver business critical applications.

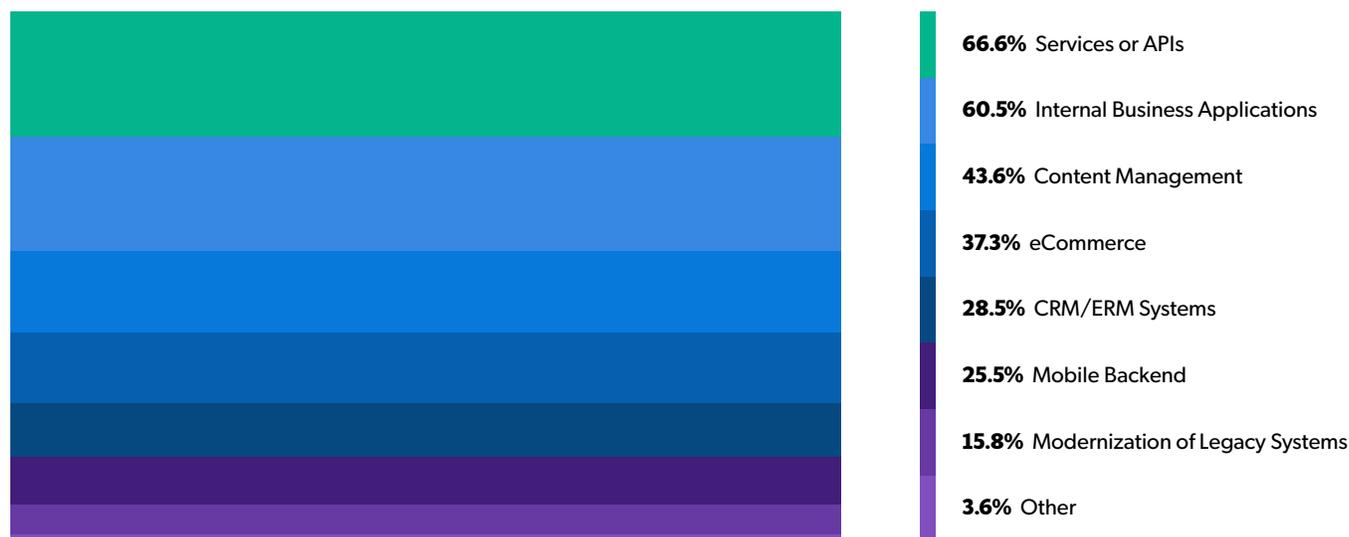
Because it is web-facing, and because these integrations can be anywhere the application can connect over the internet, developers have a choice of operating systems and cloud providers they can use for deploying their applications. The vast majority are choosing Linux operating systems, and increasingly deploying to the cloud instead of on-premises. This poses new challenges for security-conscious organizations, as well as those under regulatory requirements — which are becoming increasingly prevalent.

# What Are Teams Developing With PHP?

In our next question, we asked respondents to share the type of PHP-based applications or systems they work on. Respondents were allowed to select more than one answer choice.

The top three categories were unchanged from 2022, with over 66.6% of respondents working on services or APIs, and over 60.5% working on internal business applications. CMS systems were again third, at 46.6%.

## What Types of PHP Applications Do You Build or Deploy?



*The top development targets for PHP continue to be APIs, internal business applications, and Content Management Systems, with 60 - 67% of respondents indicating they are building APIs and internal business applications. The focus on web APIs is to be expected, as PHP's original target was web applications, and it has always "spoken HTTP" at a fundamental level. The fact that so many organizations are building internal business applications speaks to the ability of PHP to adapt quickly in order to produce business value.*

# Top Application Integration Categories

In this year's survey, to better understand the applications and systems that PHP developers are working with, we asked respondents to share the types of systems their PHP applications and systems integrate with. Respondents were able to select multiple options.

The results showed that most respondents integrate their PHP application with a variety of systems, including databases, APIs, and middleware. Relational databases were the top option at over 74.6%, with web APIs in second at over 68.3%. Filesystems rounded out the top three at over 55.1% of respondents.

## What Kinds of Systems Does Your PHP Application Integrate With?



*The acronym LAMP stands for "Linux-Apache-MySQL-PHP", and describes the most common stack used by web developers through the 2000s and into the 2010s. Developers using a LAMP stack were able to quickly create "CRUD" (Create-Read-Update-Delete) applications for managing business data, and these were typically backed by a MySQL database.*

*Today, relational databases are still the bread and butter of PHP developers, with fully three quarters of respondents indicating they use one. That said, the breadth of technology with which PHP developers now integrate is breath-taking, ranging from web APIs to key-value storage, cloud storage, message queues, search services, and more. PHP clearly allows managing the complexity of modern applications and infrastructure architectures.*

# Top Operating Systems For PHP Teams

In our next question, we asked respondents to share the operating system(s) they use for their PHP-based applications and systems. Unfortunately, the survey only allowed respondents to select a single option. Consequently, the results do not show a true representation of the total number of operating systems used.

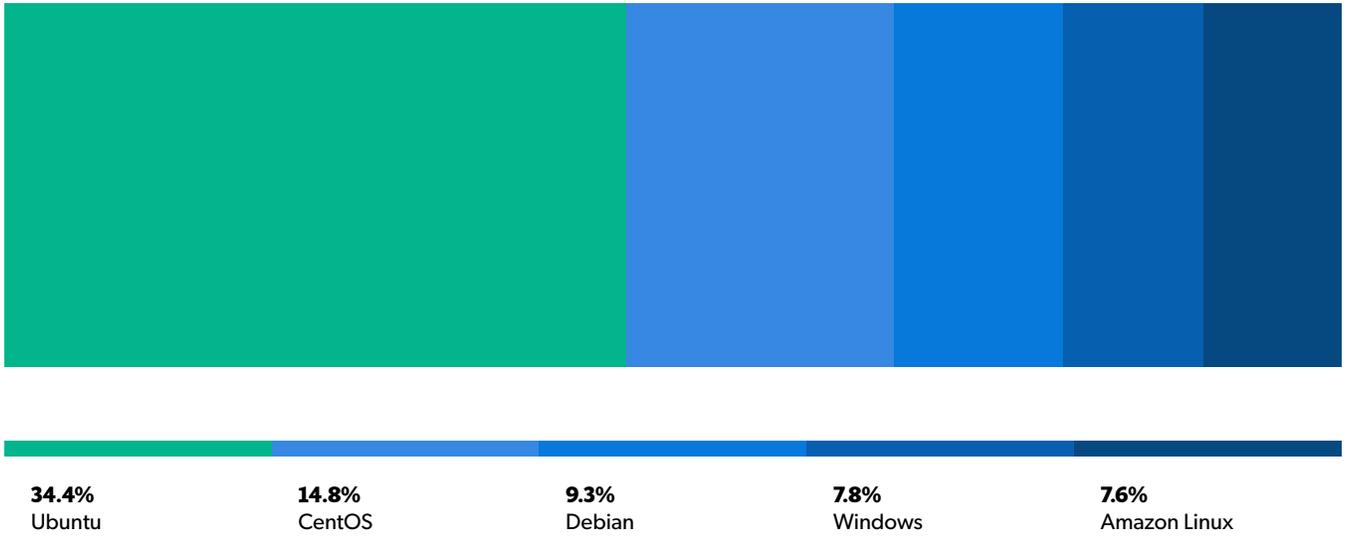
That said, we think the results showcase the perceived “primary” operating system that teams use with their PHP applications and systems. With that in mind, Ubuntu ranked first among responses at 29.5%, with CentOS (14.9%) and Debian (9.8%) in 2nd and 3rd, respectively.

## On Which Operating System(s) Do You Deploy Your PHP Applications?

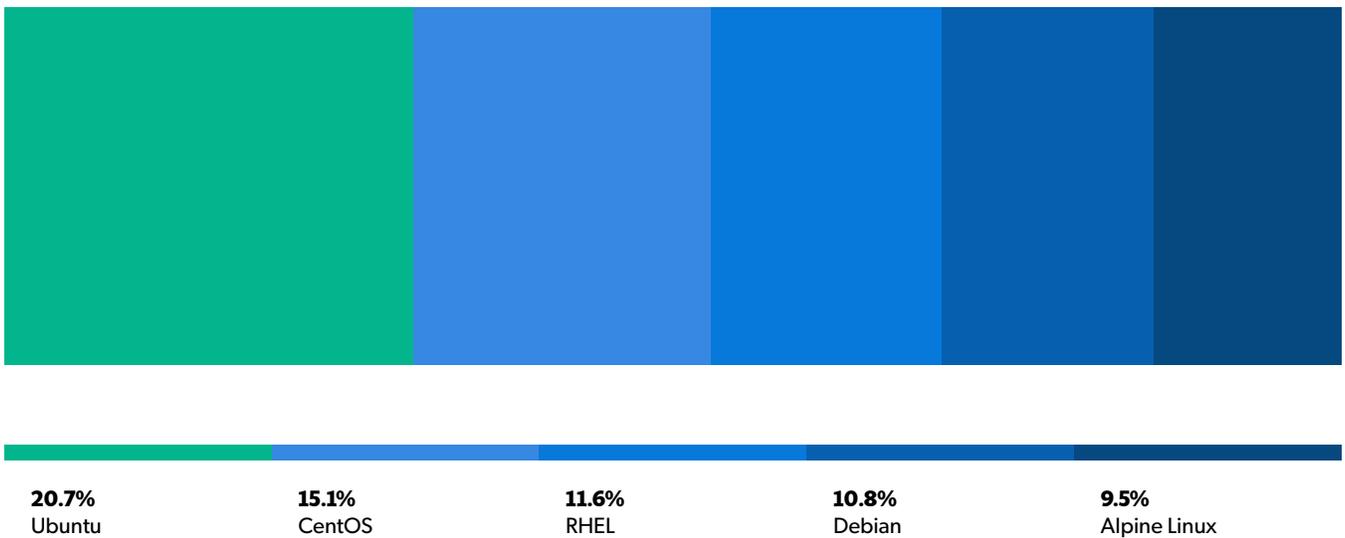


Looking at operating system choice by company size, we found some interesting results. Ubuntu was still the top operating system in both categories, however Ubuntu usage in large companies was significantly lower than in small companies. It's also worth noting that Red Hat Enterprise Linux breaks into the top five operating systems for PHP teams, with 11.69% of companies with over 100 employees using RHEL for their PHP applications.

## Under 100 Employees



## Over 100 Employees



# Where Are Teams Deploying Their PHP Applications?

In the next question, we asked respondents to share where they are deploying their PHP applications. Respondents were allowed to select multiple options.

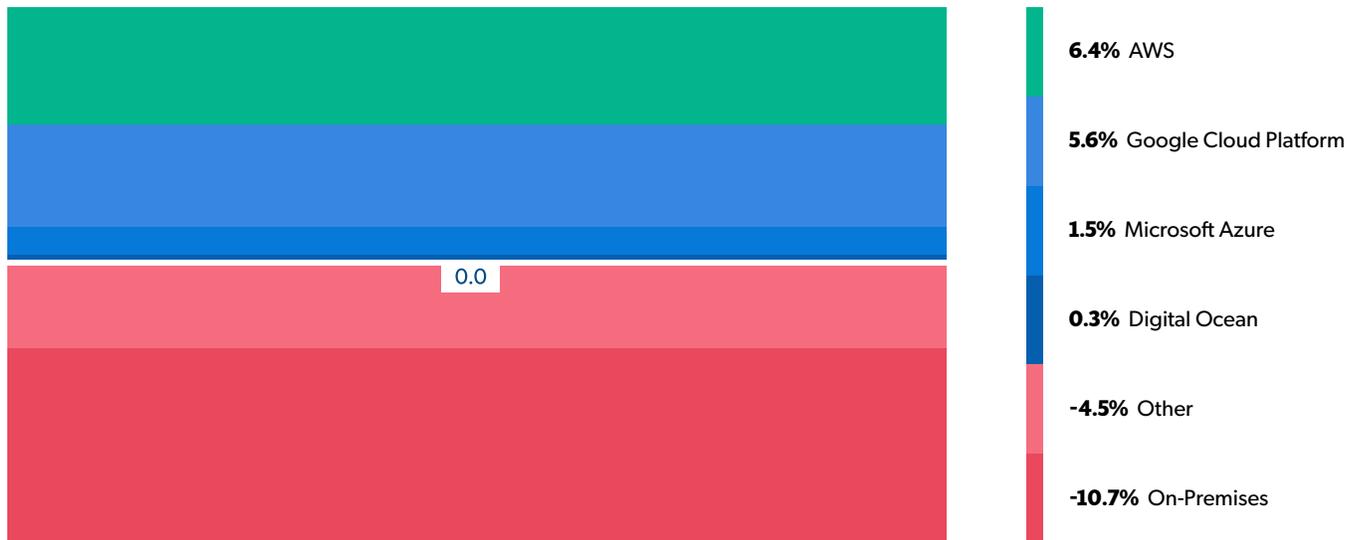
The top three deployment options were AWS (46%), on-premises (36.8%), and Google Cloud Platform (19.5%). Among respondents who selected "Other", VPS services were the most commonly provided response.

## Where Are You Deploying Your PHP Applications?



The results, when compared to 2022 results, showed a slight change in how teams are deploying their PHP applications, with AWS overtaking on-premises as the top option, and other cloud deployment options also showing growth. On-premises deployments dipped by over 10% year over year, while AWS and Google Cloud Platform gained 6% and 5%, respectively.

## Year Over Year Percentage Change



*We have noticed a trend year-on-year of organizations moving from self-hosted/on-premise solutions to cloud deployment. Across the board, each cloud platform we have tracked has shown increases this year.*

*Some of this may be attributed to the costs of self-hosting, where things like managing backups, mitigating disaster recovery, and providing geographic distribution of applications can stretch an organizations capabilities. In other cases, the various cloud platforms provide deployment and management ease-of-use that can have outsized return on investment.*

# Top Web Servers For PHP Teams

In our next question, we asked respondents to share information on the web server they use for their PHP applications and systems. Respondents were given the choice to select multiple options, as applicable.

Our survey found Apache and nginx as far and away the top options, with 57.3% and 56.5% representation, respectively.

## Which Web Server(s) Do You Use For Your PHP Applications?



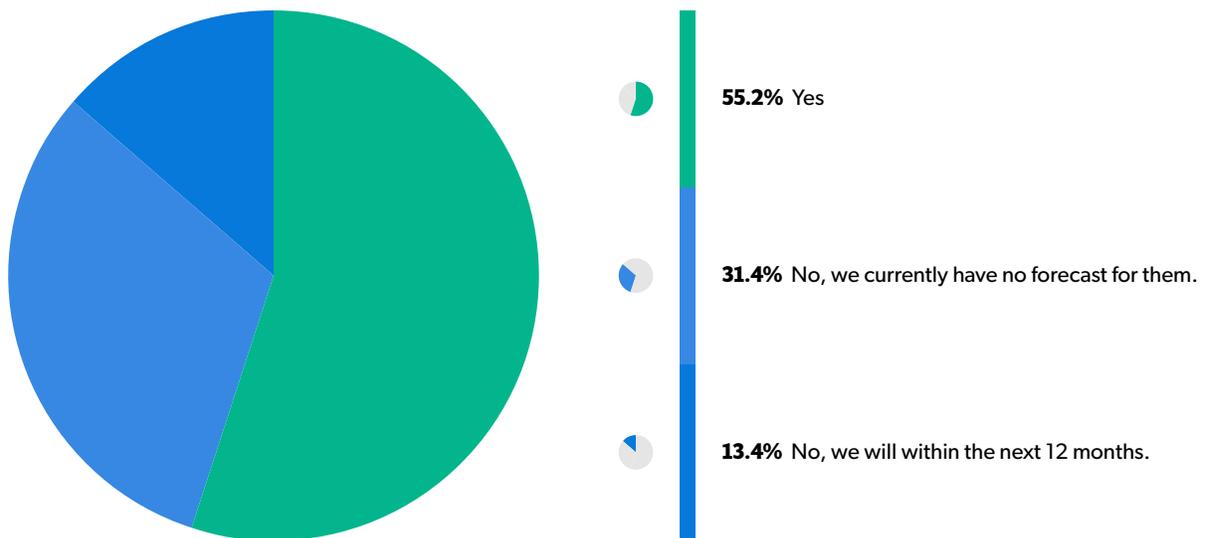
While Apache did maintain the #1 spot year over year, the percentage of users selecting Apache went down by nearly 20%. Lighttpd, LiteSpeed, and Caddy all experienced year over year increases, though they didn't represent a significant percentage of overall responses.

# Compliance Requirements

In our next question, we asked whether or not respondents' PHP applications have regulatory or industry compliance requirements.

Our survey found that the majority (55.2%) of respondents now operate applications with these types of compliance requirements. This marks a slight uptick from the 50% who noted compliance requirements for their applications in 2022. 13.4% noted that they will have compliance requirements within the next calendar year, so we expect to see the number reporting compliance requirements to increase again in 2024.

## Do Your PHP Applications Have Regulatory or Industry Compliance Requirements?



# PHP DEVELOPMENT & MAINTENANCE

In an ideal world, engineering teams would ship software free of bugs, and focus entirely on feature development that drives business success. No team is capable of this, however, which means engineering time is always split between maintenance and new functionality. Maintenance itself can take several forms: fixing reported issues, improving performance, addressing security concerns, updating dependencies, or even making the code easier to understand and maintain via refactoring. PHP developers have a wealth of tools available to them today, ranging from powerful integrated development environments to debuggers, profiling tools, and application monitoring.

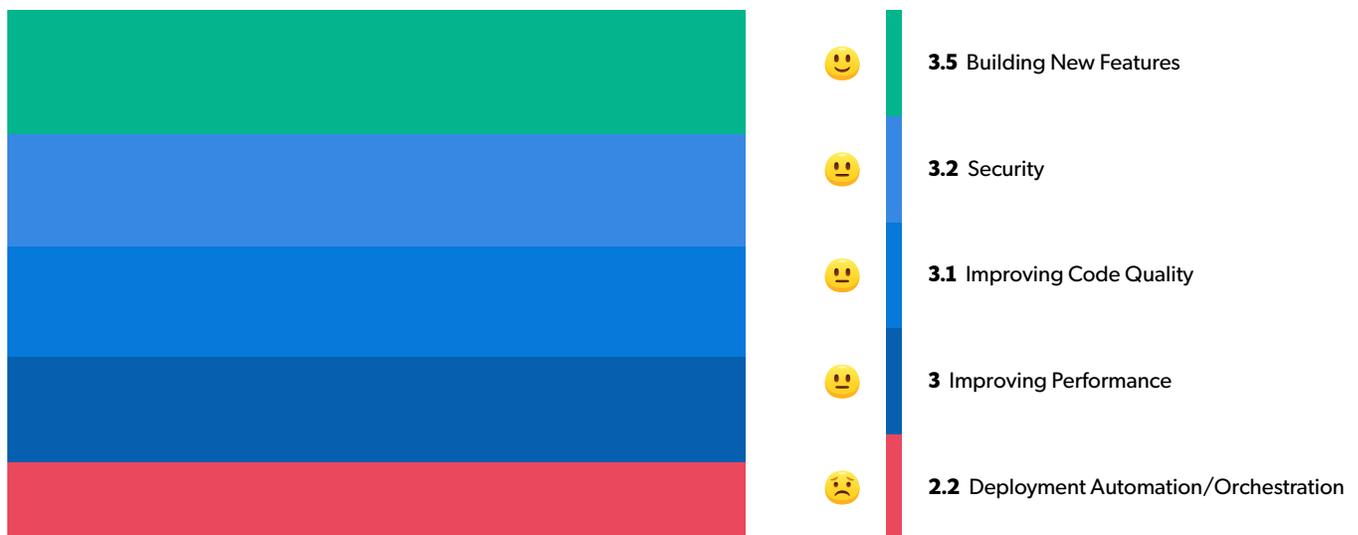
In our next set of questions, we set out to discover what PHP teams are working on, how much time they spend on different task types, and how they discover production issues.

# Top Development Priorities For PHP Teams

In our next question, we asked respondents to rank their current development priorities on a scale from one to five, with five being the highest priority.

Similar to our 2022 results, our survey found “building new features” as the highest priority, followed by “security” as the second highest priority. “Improving code quality” and “improving performance” flipped spots, with respondents viewing the former as a higher priority. “Deployment automation / orchestration” again rated as the lowest priority for PHP teams.

## What Are the Top Priorities For Your PHP Development Team in 2023?



*Deployment automation and orchestration are often seen as a task done once and then left alone, and this is reflected in our survey findings. Often, however, we find that automation can be the reason we are able to accomplish (or fail in) our other priorities. With automation, we can push our new features and security patches quickly and confidently. Automation also allows us to push changes that improve performance — and quickly revert them if we measure decreases.*

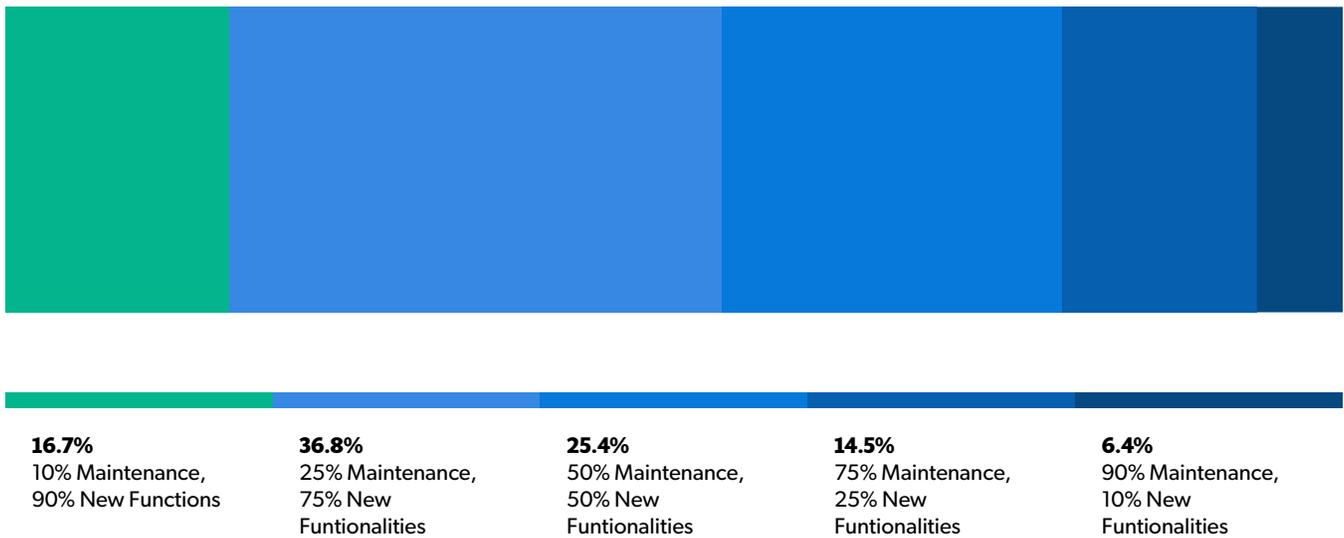
*One aspect we have also noted over time: the ability to run QA tooling in our automated CI systems has led to an increased focus on code quality in the PHP ecosystem.*

# Maintenance vs. Feature Development

Next, we asked respondents to share information on their development priorities, specifically the balance of time spent maintaining and troubleshooting their applications vs. the amount of time spent developing new functionalities.

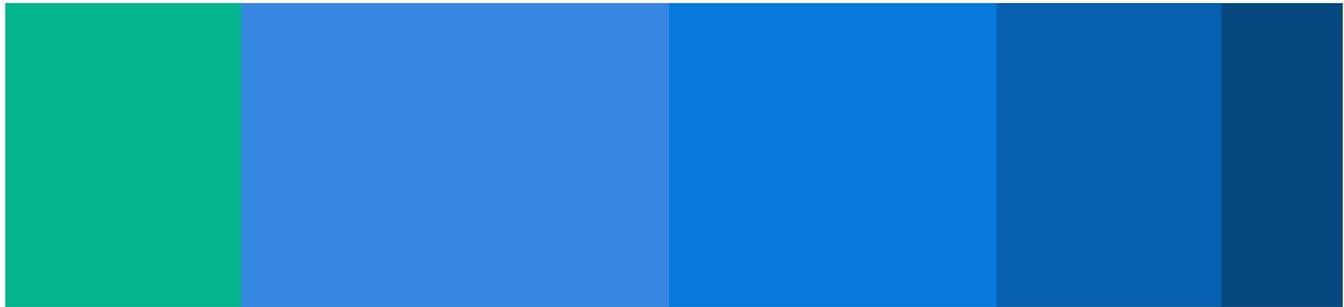
Our survey found over 36.8% of respondents spending 25% of their time on maintenance, and 75% on developing new features and functionalities. A 50/50 split was the next highest selection at 25.4%. Overall, 46.4% noted that their teams are spending at least 50% of their time on maintenance and troubleshooting.

## How Much of Your Team’s Time is Spent on Maintenance and Bug/Issue Resolution vs. Developing New Functionalities?



When looking at responses by company size, companies with over 100 employees had a 4% higher number of teams spending more than half their time on maintenance and troubleshooting.

## Companies Over 100 Employees



<b>17.6%</b> 10% Maintenance, 90% New Functions	<b>31.9%</b> 25% Maintenance, 75% New Funtionalities	<b>24.5%</b> 50% Maintenance, 50% New Funtionalities	<b>16.8%</b> 75% Maintenance, 25% New Funtionalities	<b>9%</b> 90% Maintenance, 10% New Funtionalities
---	---	---	---	--

*With close to half of respondents indicating they are spending 50% or more time on maintenance, there is clearly room for improvement when developing PHP applications. A common theme observed in the PHP ecosystem the past few years is that the pace of change within the PHP language itself is leading to issues managing PHP upgrades, particularly with regards to feature deprecations and subtle language changes that can lead to application errors.*

*One solution for this is to use long-term support editions of the language to delay migrations; however, this may not address things such as lack of security fixes in versions of PHP libraries on which your applications depend. As always, investing in CI/CD pipelines can help you identify and issues early in your development process; Zend offers services to help you implement these.*

# Finding Issues in Production

Our next question asked respondents to weigh in with how they solve issues in their production applications. Respondents were able to select multiple answer choices.

Overall, the top reported method for teams was “application logs” at 77%, followed by user reports at 66%. APM tools finished in third at 42%. Compared to 2022, we saw a slight increase in those choosing “user reports” and a slight decrease in those choosing “application logs.”

## How Do You Discover Issues in Your Production Application?



**77.7%**  
Application Logs

**66.2%**  
User Reports

**42.5%**  
Application Performance  
Monitoring (APM) Tools

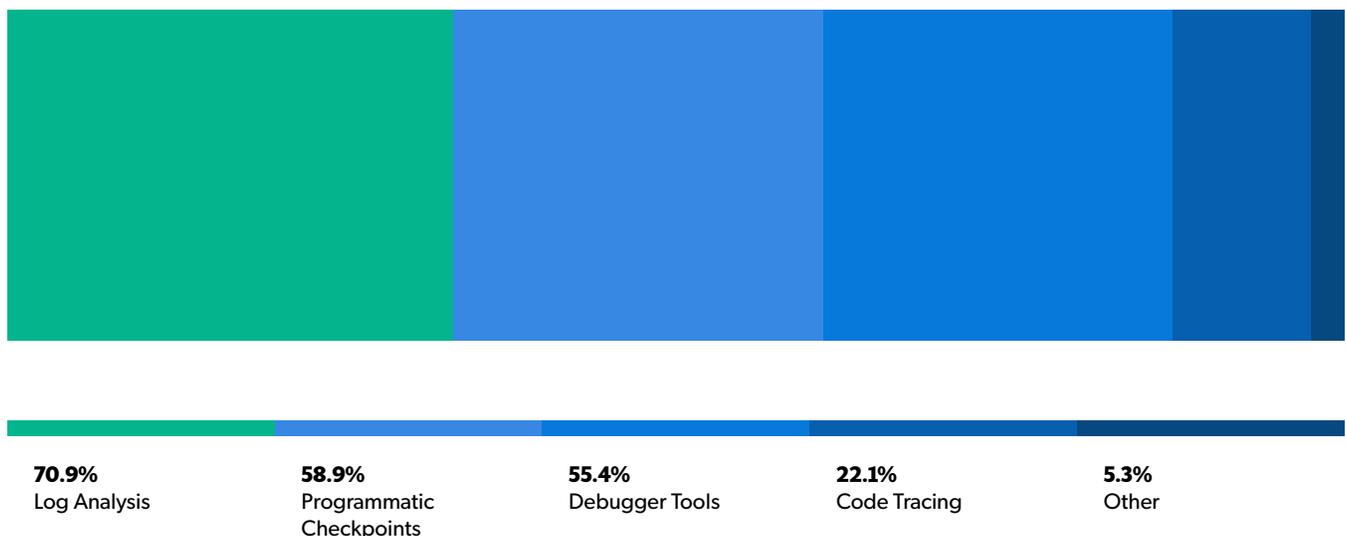
**5.8%**  
Other

# Solving Issues in Production

In our next question, we asked how respondents identify and resolve issues in their production applications. Respondents were able to select multiple answer choices.

Matching the results from the previous question, the top selected option was “log analysis” at nearly 70.9% of responses. Moving down the list, 58.9% of respondents reported “programmatic checkpoints”, 55.4% reported “debugger tools,” and 22.1% reported “code tracing.”

## How Do Your Developers Identify and Resolve Production Issues?



*Solving issues in production can be hugely daunting, as it is often difficult to precisely recreate the full context that led to an issue. Logging is often touted as a best practice, but analyzing logs to understand context requires a huge amount of skill and experience.*

*Using debugging tools in production — whether that’s a dedicated debugger such as XDebug or programmatic checkpoints — can often lead to decreased performance, and occasionally to security issues (due to information disclosure). We recommend using debuggers only in development — which unfortunately still requires identifying the context leading to an issue first. These shortcomings are precisely why Zend has invested in Code Tracing tools for Zend Server and ZendPHP, as they provide the full error context, allowing developers to reproduce an issue, as well as identify root causes.*

# CONTAINERIZATION & ORCHESTRATION

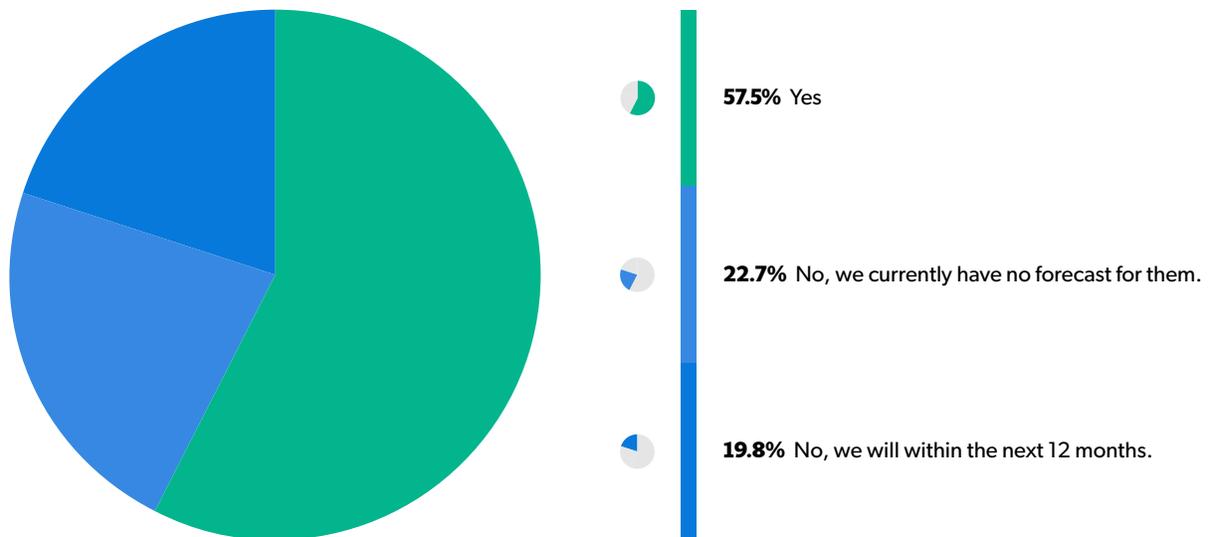
Containerized deployments are quickly becoming the norm for PHP applications. Since containers provide repeatable, idempotent deployment, they are a perfect solution particularly when an application requires autoscaling. Other benefits include the ability to develop applications in conditions closely mimicking production, the ability to model all infrastructure integrations, and the ability to lock down networking between services in ways that are difficult to accomplish in traditional hosted environments.

# Container Usage

Next, we asked respondents to share their status and plans for containerizing their PHP applications.

Our survey found over 57.5% of respondents currently using containerization technologies, with an additional 19.8% planning on using container technologies within the next year. Only 22.7% noted they were not using container technologies, and had no plans to do so within the next year.

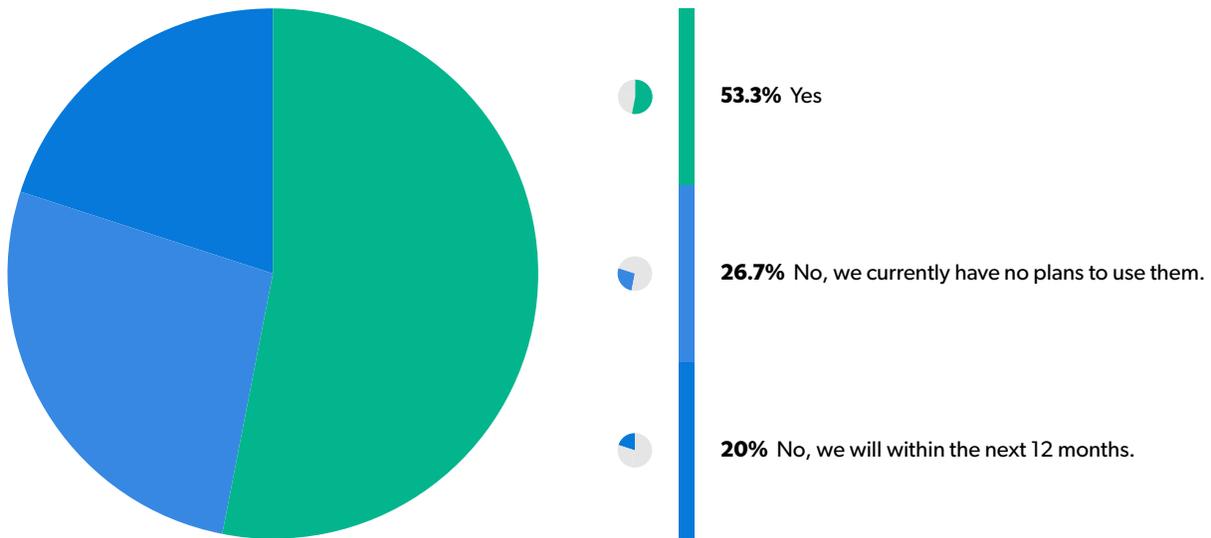
## Are You Currently Using Any Containerization Technologies?



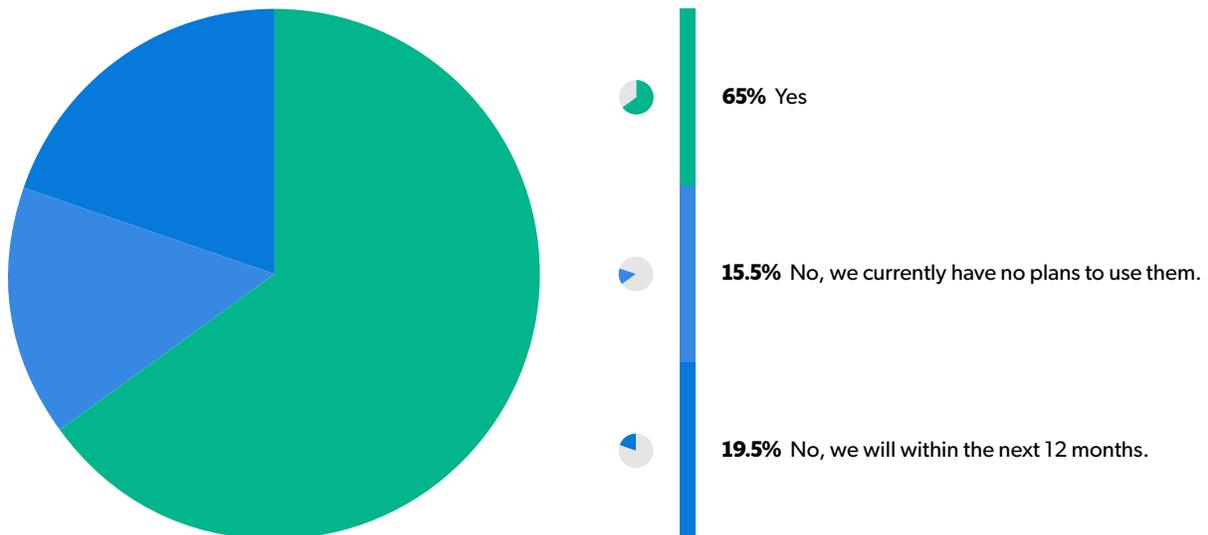
Compared to last year, there were fewer respondents noting plans to use container technologies within the next 12 months, with nearly 20% in 2023 compared to 47% in 2022. However, in 2023 we noted significantly more teams using container technologies, with over 57% in 2023 vs. 30% in 2022.

Looking at container adoption by organization size, we found 65% of companies with over 100 employees currently using container technologies. Companies with under 100 people were over 10 points lower at 53.3%.

## Companies With Under 100 Employees



## Companies With Over 100 Employees



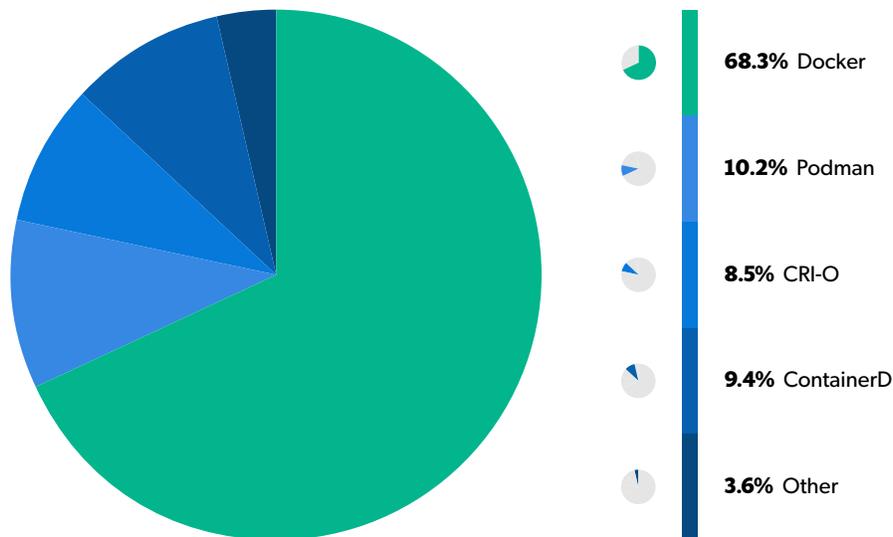
*In 2022, we observed 47% of respondents indicating they would be adopting containers in the coming year, and 30% already using them. In 2023, we observed 20% indicating they would adopt in the coming year, and 57% already using them. While we understand that the respondents are not the same set, we find it interesting that each of these numbers changed by 27% year on year; the takeaway is that organizations are creating and being successful with plans to containerize their PHP applications.*

# Top Container Technologies

In our next question, we asked teams to share the container technologies they use with their PHP applications.

Our survey found Docker as the most popular container technology with over 68.3% of responses, followed by Podman (10.2%), ContainerD (9.4%), CRI-O (8.5%) and Other at 3.6%.

## Which of the Following, If Any, Containerization Technologies Do You Use?



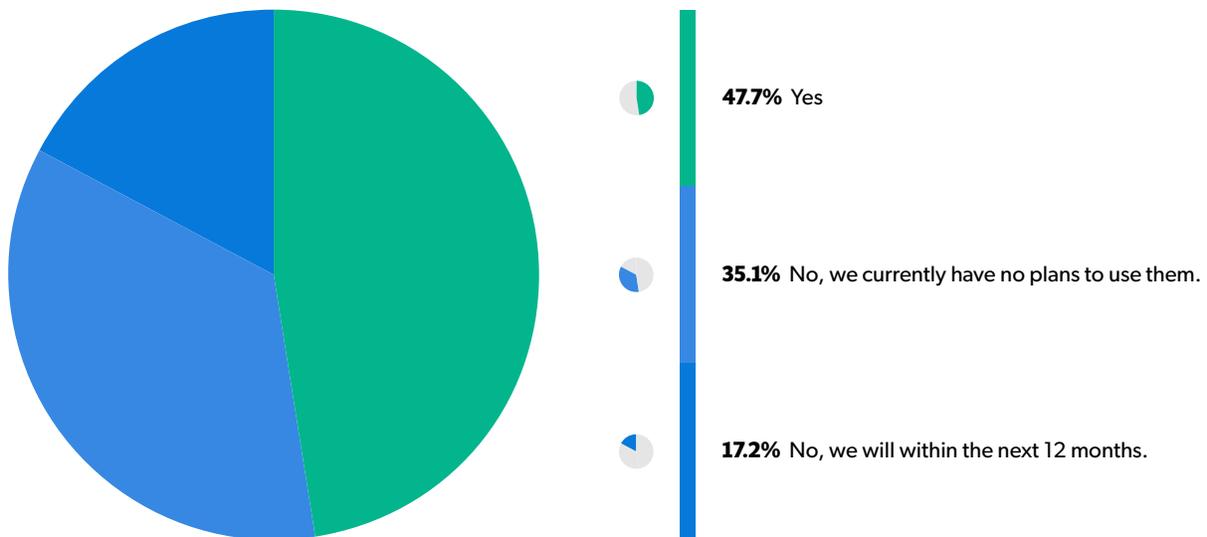
*While organizations have a variety of container technologies to choose from (in large part due to the work of the Cloud Native Computing Foundation and its partners), Docker is clearly the technology of choice when developing containerized applications for PHP.*

# Orchestration Usage

Moving from containers to orchestration, our next question asked respondents to share their status in the adoption of orchestration technologies.

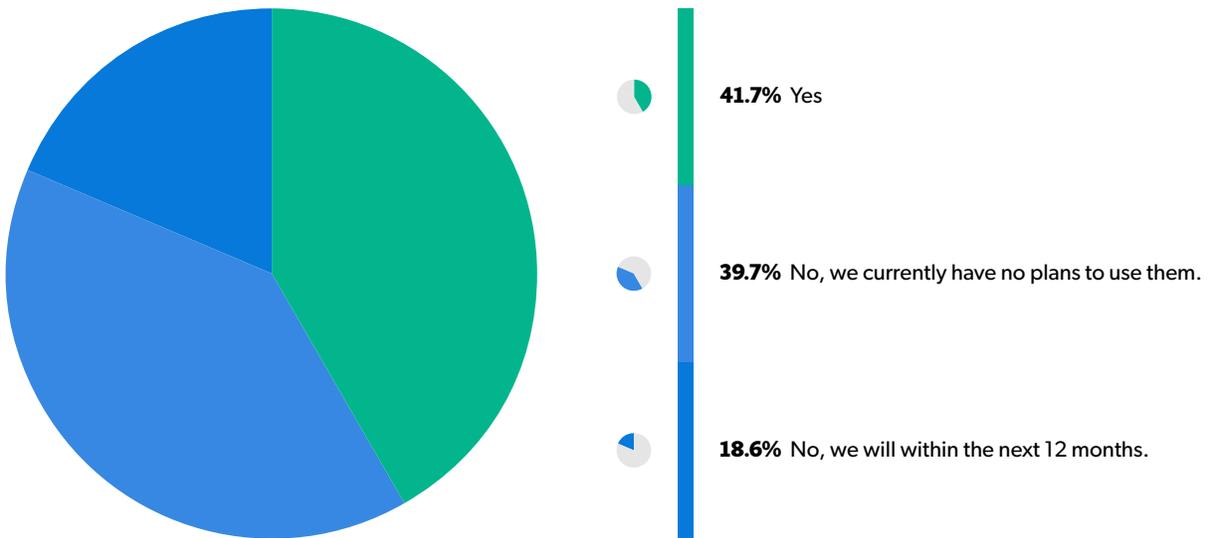
Our survey found over 47.7% of teams using orchestration technologies, with an additional 17.2% planning on using them within the next 12 months. Only 35.1% noted no plans to use orchestration technologies.

## Are You Currently Using Any Orchestration Technologies?

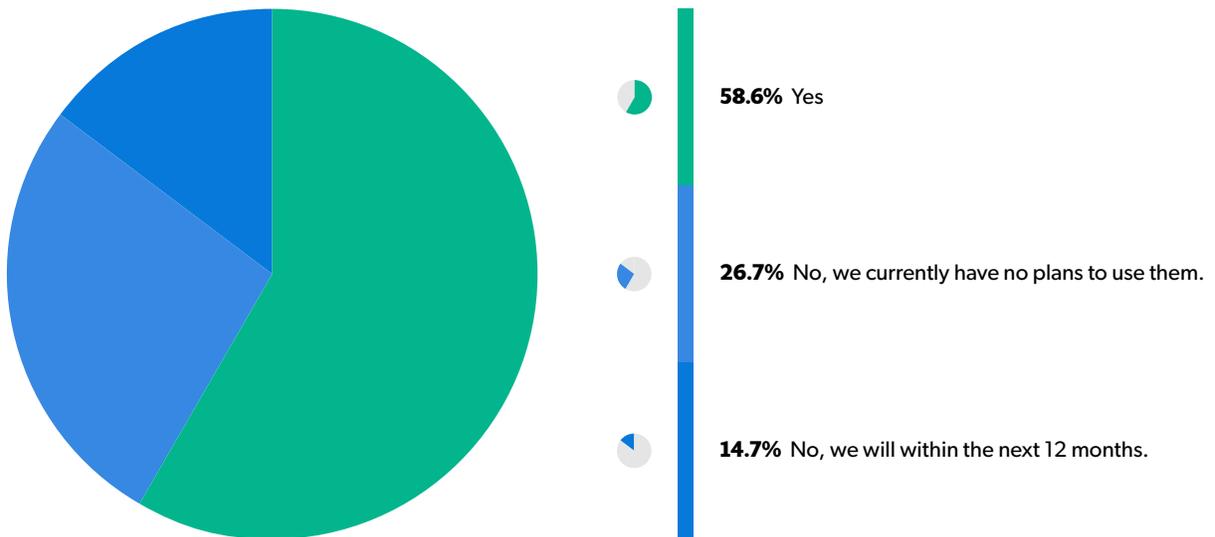


When looking at these results by company size, we found that companies with over 100 employees were much more likely than companies with under 100 employees to be using orchestration technologies, with 58.6% vs 41.7% respectively.

## Companies With Under 100 Employees



## Companies With Over 100 Employees



# Top Orchestration Technologies

Following that question, we asked teams to share the orchestration technologies they use. Respondents were given the option to select multiple technologies.

Our survey found Kubernetes as the top orchestration technology with 44.8% of responses, followed by Ansible (31.9%), Terraform (27.85%), AWS Cloud Formation (25.3%), and Helm (19.4%).

## Which of the Following, If Any, Orchestration Technologies Do You Use?



*While popular cloud platforms such as AWS provide platform-specific orchestration technologies, consumers are clearly gravitating to open source solutions such as Kubernetes and Terraform, often combining them with other automation and provisioning tools such as Ansible and Puppet.*

*With the growing number of containerized applications, it's no small wonder that Kubernetes was a favorite amongst respondents, and Ansible, at number two, is often used in conjunction with Kubernetes in tools such as kubespray. Standardizing on open source solutions allows organizations to have hybrid and multi-cloud strategies that avoid vendor lock-in, and allow selecting platforms based on the value they provide for your PHP applications.*

# PHP VERSION & UPGRADE STATISTICS

PHP versions follow a three-year lifecycle, with two years of active support, and an additional year of security-only support. This lifecycle allows the language to advance on a predictable lifecycle, but also leads to year-on-year churn for organizations using the language, as they need to constantly update infrastructure and test their applications against new versions.

Compounding that this year is the fact that PHP 7 reached end of life with the end-of-life announcement of PHP 7.4 in November 2022. Organizations who have not yet adopted PHP 8 now find themselves needing to either migrate quickly, or find a commercial provider of a long-term-support edition of the language.

Each year, we ask what PHP versions our respondents use in production to get an idea of the pace at which new versions are being adopted, and how many in the ecosystem are continuing to use outdated PHP versions.

# Most Used PHP Versions

In our next question, we asked teams to share which versions of PHP they deploy as part of their PHP applications. Respondents were given the option to select multiple PHP versions.

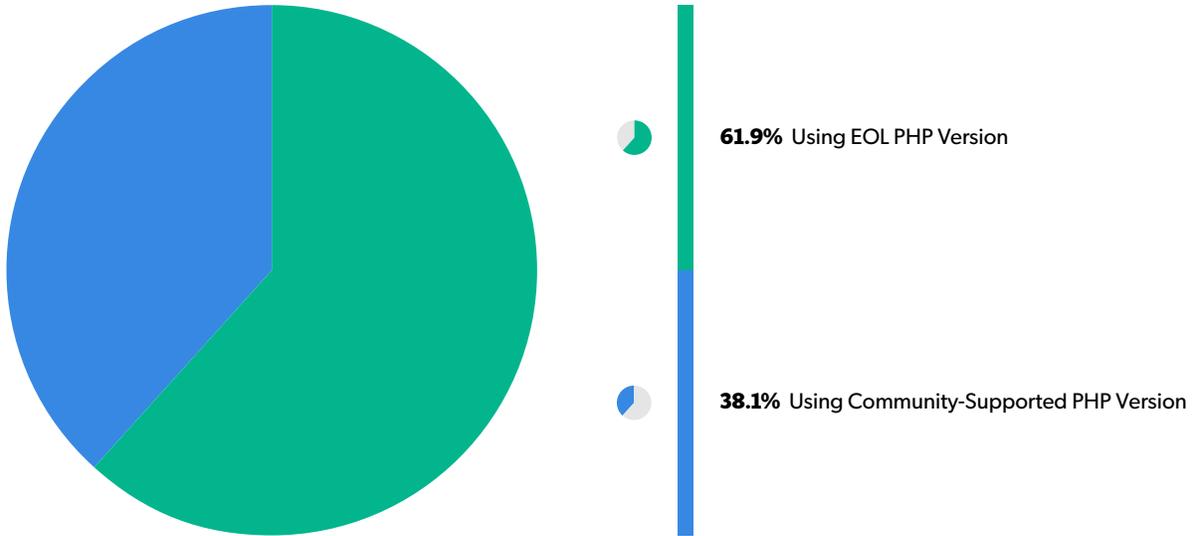
PHP 7.4, now end of life, received the most responses – with 54.2% reporting use of PHP 7.4 in their PHP applications. PHP 8.1 and PHP 8.0 were the next most popular, with 46.1% and 35.4% of responses, respectively. Compared to the 2022 survey, PHP 7.4 usage dropped from 66% to 54%, with PHP 8.0 usage also dropping (44% to 35%).

## Which Versions of PHP Do Your PHP Applications Use?

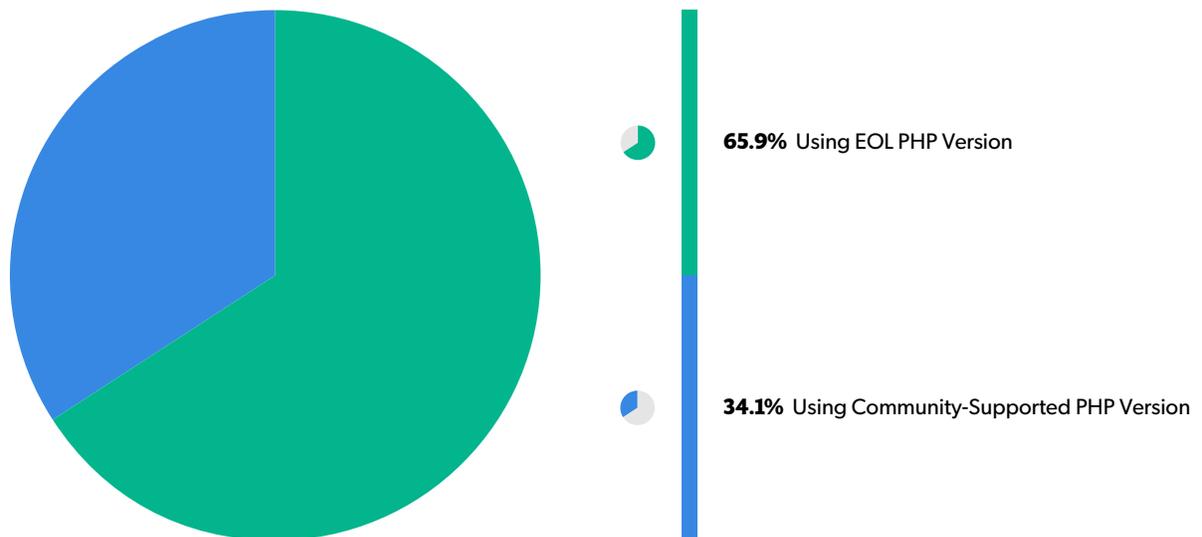


Overall, teams using end of life (EOL) PHP versions represented 61.9% of all responses. When looking at this data for companies with over 100 employees, we found that number jump up to 65.9%, while companies with under 100 employees dropped to 59.4%.

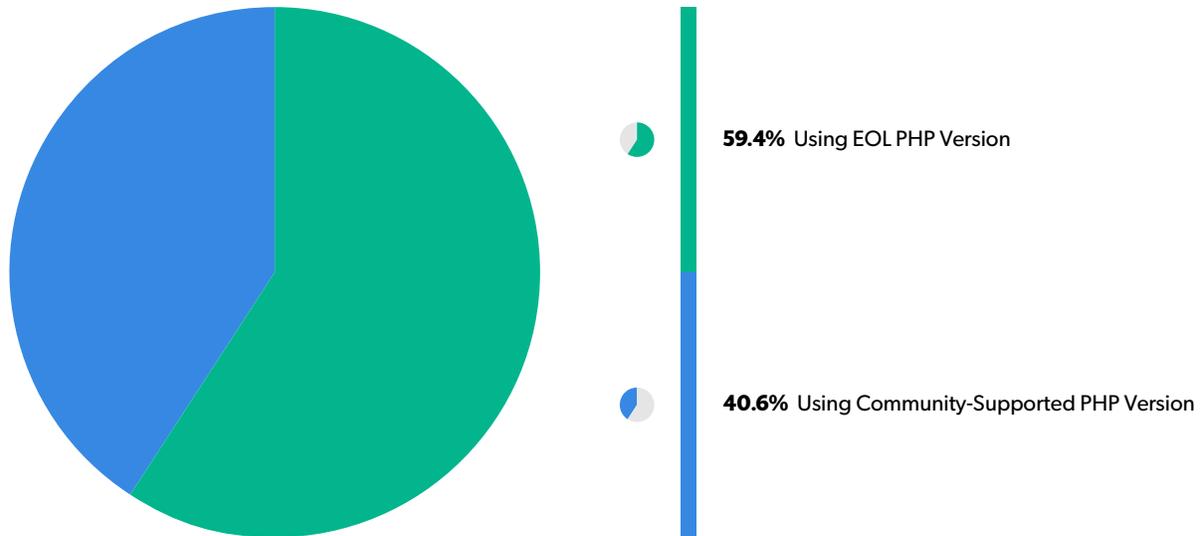
## Comparison of Results by PHP Version Community Support Status



## PHP Version Community Support Status for Companies With Over 100 Employees



## PHP Version Community Support Status for Companies With Under 100 Employees



*Three out of five respondents indicate they are still using an EOL version of PHP in production. When we asked about this in social media, many indicated they feel the pace of change in the language makes keeping up-to-date difficult.*

*One question raised by many was whether or not the language could periodically create LTS versions with longer security lifecycles (for example, a total of 6-10 years instead of 3); this would allow developers to target the LTS versions for applications with long-term lifecycles, and ease maintenance for library authors as well.*

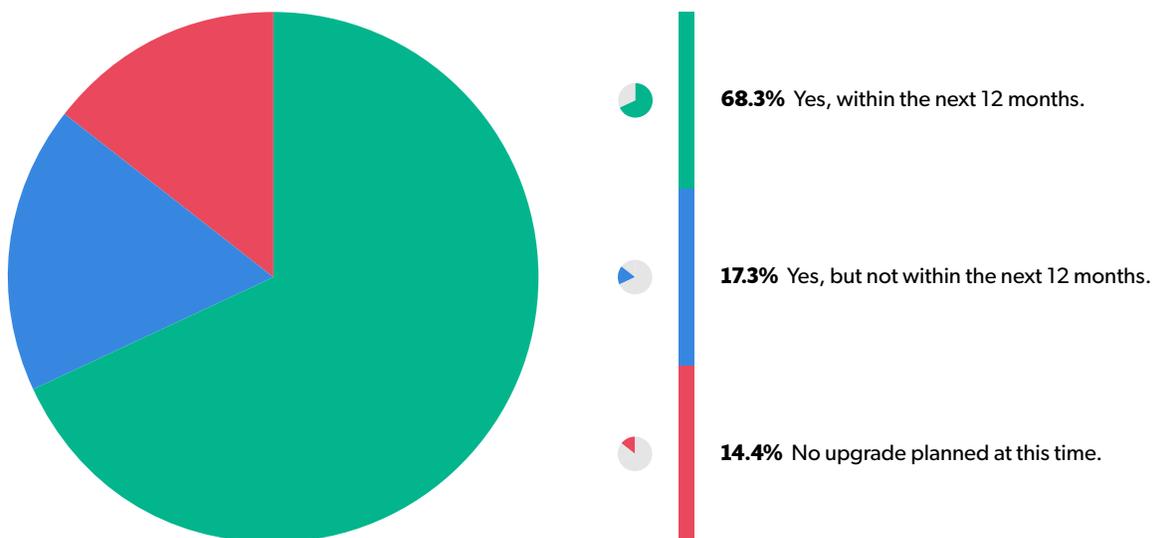
*Critics of this idea pointed out the maintenance burden on language developers and release managers, and suggested that this is rightfully the realm of commercial providers such as Zend. We anticipate this will continue to be a subject of heavy debate with no clear outcome.*

# PHP Version Upgrade Planning

In our next question, we asked respondents to share their plans for performing their next PHP upgrade or migration.

Our survey found 68.3% of respondents reporting a planned upgrade within the next 12 months, with an additional 17.3% planning an upgrade outside of that window. Only 14.4% reported not having a planned upgrade or migration.

## Are You Planning a PHP Version Upgrade or Migration Within the Next Year?



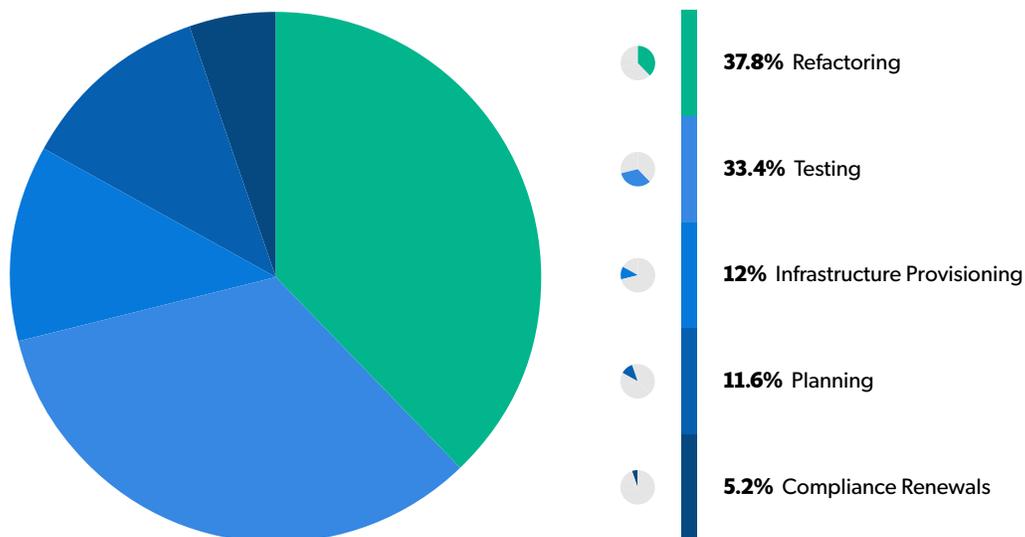
*Each year when we ask this question, we see respondents in the 60-70% range indicating a plan to perform a PHP upgrade in the coming year. However, similarly, we see 60-70% of respondents still on end-of-life versions. We anticipate some of this is due to needing to perform stepped upgrades: for example, migrating from PHP 5.6 to 7.2, then from 7.2 to 7.4, and then to a PHP 8 version. Such upgrades take significant investment in planning, testing, and infrastructure provisioning. The question, again, is if the language lifecycle is leading to these unbalanced results.*

# PHP Upgrade Challenges

Following that question, we asked teams to share the most time-consuming part of their last PHP upgrade or migration.

37.8% noted refactoring as their most time-consuming component, followed by testing at 33.4%. Infrastructure provisioning was the next most time-consuming component at 12%, followed by planning and compliance renewals at 11.6% and 5.2%, respectively.

## What Was The Most Time-Consuming Component of Your Last PHP Upgrade?



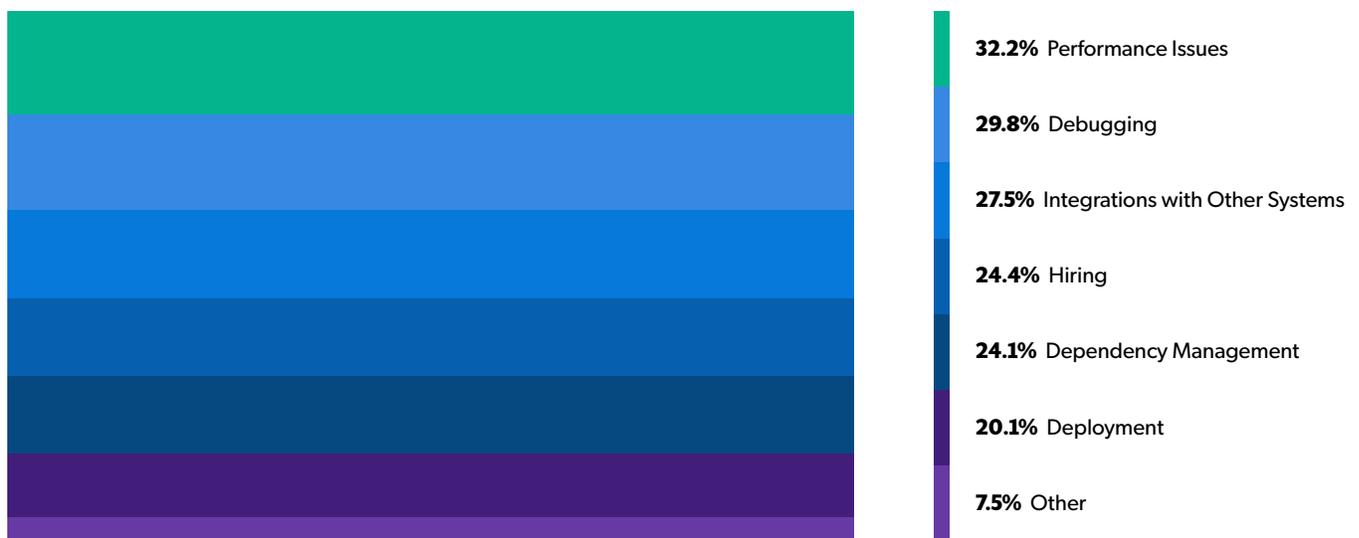
*While PHP strives to keep backwards compatibility between versions, we have observed that each version introduces subtle changes that have outsized impact on applications. Additionally, when deprecations are finally removed, they can lead to applications breaking in unexpected ways. As such, we are not surprised to see that refactoring and testing consume the bulk of time for respondents. Identifying what has changed, and how to adapt your code, while simultaneously ensuring all existing functionality works as expected are difficult tasks.*

# Top Challenges For PHP Teams

Continuing the theme of hurdles for PHP teams, the next question asked respondents to share their biggest challenge in working with PHP.

Our survey found performance issues as the most common problem with 32.2% of responses, followed by debugging at 29.8%. Integrating with other systems came in third at 27.5%, with hiring and dependency management the next most common options at 24.4% and 24.1%, respectively.

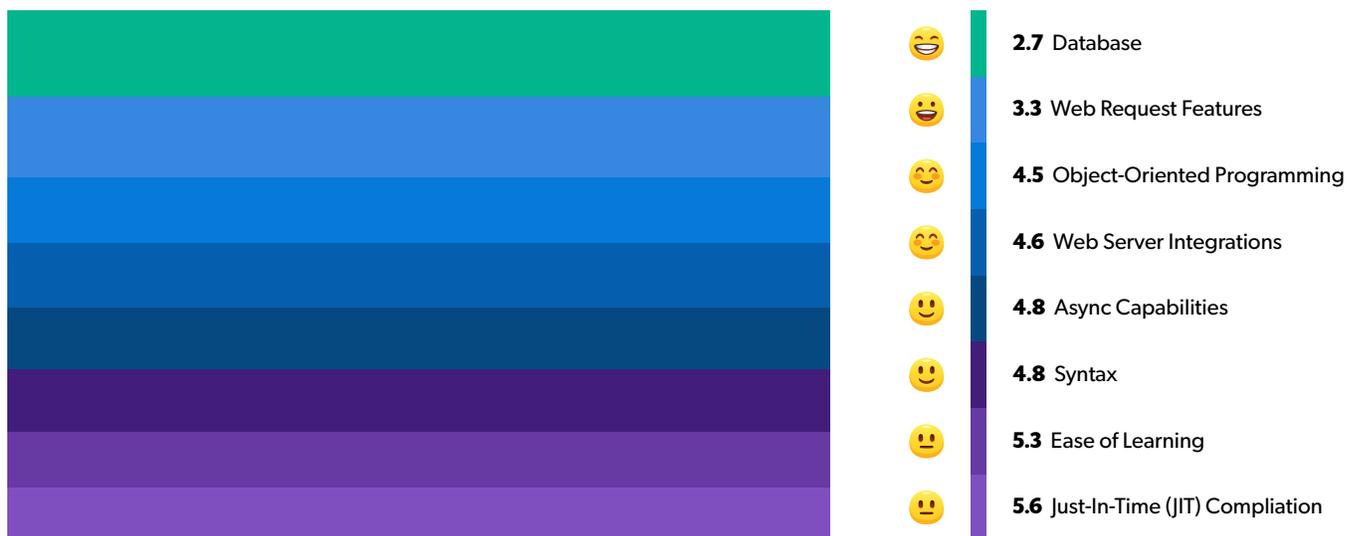
## What Are Your Biggest Challenges Around PHP?



# Most Valuable PHP Features

In our next question, we asked respondents to weigh in with what they consider to be the most important PHP features to their work. Respondents were asked to rank each feature on a scale of one to ten, with one representing the most important feature and ten representing the least important feature.

## What Features of PHP Are Most Important to You? Please Rank in Order of Importance.



With an average rating of 2.74, respondents viewed database and other service connectivity features as the most important PHP feature set, followed by web request features (3.38), OOP features (4.5), web server integrations (4.68), and async capabilities (4.84). The lowest rated feature was JIT compilation, which averaged a 5.63 rating out of 10

*It's not surprising to see database and other service connectivity features at the top of the list, especially when you consider three-quarters of respondents noted they are using a relational database with the language. Web request features are no surprise, either, considering that PHP is still far and away the favored backend language for web applications.*

*While JIT compilation is a good advancement for the language, we weren't surprised to see it as the least important feature for respondents. While it can contribute to performance improvements, the various benchmarks and tests have indicated that its impact on the vast majority of PHP applications is negligible.*

# The State of PHP

For our final question, we asked respondents to share their rating of a few key areas of PHP development. Respondents were able to rank each category from one to five, with one representing the lowest level of agreement and five representing the highest level of agreement.

Our survey found generally positive responses to all questions, people generally enjoying working with PHP, and agreeing that PHP is easy to learn. The only question that received negative sentiment was the agreement with “PHP evolving too slowly.”

## On a Scale of 1-5, With 5 Being the Highest Level of Agreement, Please Rate the Following Statements.



*We were pleased to see so many respondents indicate that they feel strongly that they enjoy working with PHP, and that it is easy to learn. This contradicts the popular narrative in the tech press that PHP is despised by its developers and a dying language. With upwards of 5.5 million developers, and powering up to 80% of the internet, the language is clearly widely used and deployed, and the survey results paint a picture of largely satisfied developers and organizations.*

*Despite some clear misgivings about the length of the PHP version lifecycle, respondents are also clearly okay with the pace of change overall; the results indicate that they neither think it is too slow nor too fast. As noted in previous sections, the language has made a number of strides in the past half decade towards improvements that have an effect on application maintainability and correctness, and the language has a strong ecosystem of QA tooling. We hope this positive trend continues for many years to come.*

# FINAL THOUGHTS

PHP's ease of learning acts as a gateway for developers to adopt the language, and the rich syntax and features, particularly from the PHP 7 and PHP 8 versions, make developers happy and productive when using the language. In particular, its focus on integrating with a wide array of technology, from relational databases to message queues, allows developers to use PHP to glue together disparate systems within an organization. These factors contribute to its success in building applications and web APIs that power businesses of all sizes, from personal blogs to the health industry, financial institutions, and governments.

If your organization is using PHP, and you're not yet using containers or orchestration techniques, we recommend investing in these areas. While they pose challenges to learn and execute well, they will lead to more stable and repeatable deployments, providing insurance against rolling out bugs to production. We also recommend taking time to research and invest in quality monitoring solutions to assist in identifying production issues, as well as their root causes. Combining these with infrastructure-as-code and a good continuous integration pipeline, you will also be arming your organization to adapt quickly to PHP updates.

We look forward to seeing what changes occur in the PHP ecosystem next year!

# About Zend

Zend by Perforce helps organizations use enterprise PHP to build innovative web and mobile solutions and modernize existing applications. Used by multiple Fortune 100 companies, our proven enterprise PHP offerings include secure, fully-supported PHP runtimes, software infrastructure, tools, certifications, and enterprise long-term support for PHP 7.1, 7.2, 7.3, and 7.4.

For more information, visit [www.zend.com](http://www.zend.com).

## Learn More About Our PHP LTS Options

[SEE LTS OPTIONS](#)

## Learn More About Our Professional Services

[SEE PROFESSIONAL SERVICES](#)