NOVEMBER 2022

NGINX State of App and API Delivery Report

Created by

/DATA NGINX Part of F5

Can I share data from this report?

1. License Grant

This report is licensed under the Creative Commons Attribution-NoDerivatives Licence 4.0 (International) . Put simply, subject to the terms and conditions of this license, you are free to:

Share — You can reproduce the report or incorporate parts of the report into one or more documents or publications, for commercial and non-commercial purposes.

Under the following conditions:

Attribution — You must give appropriate credit to SlashData™ , and to NGINX, as sponsors of this report, and indicate if changes were made. In that case, you may do so in any reasonable manner, but not in any way that suggests that SlashData™ endorses you or your use.

NoDerivatives — you cannot remix or transform the content of the report. You may not distribute modified content.

2. Limitation of Liability

SlashData™, believes the statements contained in this publication to be based upon information that we consider reliable, but we do not represent that it is accurate or complete and it should not be relied upon as such. Opinions expressed are current opinions as of the date appearing in this publication only and the information, including the opinions contained herein, are subject to change without notice. Use of this publication by any third party for whatever purpose should not and does not absolve such third party from using due diligence in verifying the publication's contents. SlashData™ disclaims all implied warranties, including, without limitation, warranties of merchantability or fitness for a particular purpose.

SlashData™, its affiliates, and representatives shall have no liability for any direct, incidental, special, or consequential damages or lost profits, if any, suffered by any third party as a result of decisions made, or not made, or actions taken, or not taken, based on this publication.

The analyst of the developer economy | formerly known as VisionMobile

ABOUT THE AUTHORS

Sam Coates

Data Storyteller

Sam is a researcher with a passion for data analysis and exploring what drives change. He holds a PhD in Physics and is the author of over 20 scientific papers, which have been published in internationally recognised journals.

✉ sam.coates@slashdata.co

Konstantinos Korakitis

Director of Research

Konstantinos heads the Research Product team at SlashData and is responsible for all syndicated research products and custom research projects. With more than 10 years of experience as an engineer, consultant and manager, he oversees research planning, survey design, data analysis, insights generation and research operations.

✉ konstantinos@slashdata.co

TABLE OF CONTENTS

KEY INSIGHTS

- 31% of all development roles also identify with leadership roles. →
- 44% of employees at large enterprises have nothing to do with security. →
- The largest issue faced by the NGINX community is a lack of technical skills. →
- 77% of the respondents who use a container orchestration tool are using a Kubernetes-based one. →
- Scalability is the number one motivation for Kubernetes adoption. →
- The top 3 code deployment environments are public cloud, web client/front-end, and on-premises servers. →

INTRODUCTION

In this report, commissioned by F5 NGINX and authored by SlashData, we explore the current state of the NGINX community. We provide an overview of the profiles of respondents to the survey in terms of their geographic location, role, and size of their organization, before focusing on their use cases and the challenges they face in application (app) and API delivery projects. We then dive deeper into apps and APIs, looking at how respondents' companies approach APIs. We also explore how important security, scalability, and observability features are in app and API delivery projects.

Moving forward, we look at the technology choices and development environments of NGINX community members, with a focus on their workloads, Kubernetes adoption/maturity, where their code is run, and attitudes towards open source software. We examine how role and organization size affects each of these topics, and compare the profiles of those with low and very high workloads. Finally, we take a look at which management/security and monitoring/observability tools the community uses, discuss cross-usage, and explore the differences between the profiles of those who use NGINX and those who don't.

The findings in this report are based on the 2022 NGINX Survey, which was designed and hosted by SlashData and fielded by F5 NGINX between August and September 2022, with more than 2,000 respondents worldwide.

PROFILE OF SURVEY RESPONDENTS

01

Roles

In this first chapter, we break down the profiles of the respondents to the 2022 NGINX Survey with a focus on their roles, the size of the organization they work for, their use cases, and the challenges they face in their API and app delivery projects.

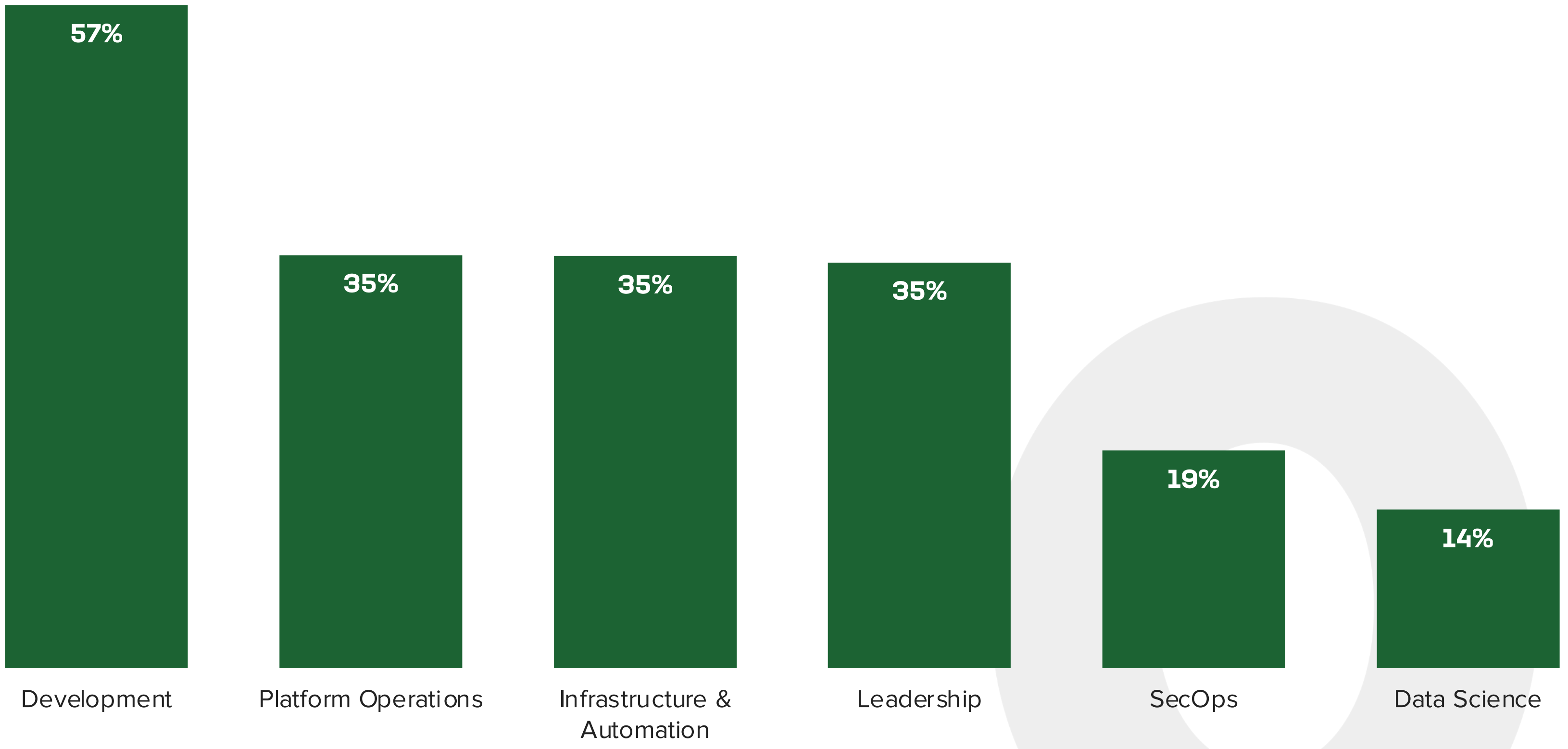
To start, let's look at roles: we asked respondents to select the types of roles that described them best, with the provided options ranging from software developer to CEO. These roles can be grouped into 6 categories: Development, Platform Operations (Platform Ops), Infrastructure & Automation, Leadership, Security Operations (SecOps), and Data Science. For a full breakdown of which roles fit into these categories, see the appendix.

While it may be no surprise that Development teams represent the largest share of the audience (57%), the relatively high percentage of those involved in leadership roles indicates that the NGINX community is quite mature in terms of responsibility. In fact, 31% of all Development respondents also identify with leadership roles, suggesting an impressive share of respondents are able to develop at the front line while also maintaining an overarching leadership mindset to their development practices.

Over half of respondents identify with roles associated with development teams

% of respondents (n=2,092)

Roles within the NGINX community



Organization Size

When looking at the sizes of organizations that professional respondents work for, we find:

- 13% are freelancers,
- 27% work for small businesses (2–50 employees),
- 30% for medium-sized businesses (51–1,000 employees),
- 30% for large enterprises (more than 1,000 employees).

Looking at roles by organization size, we find that the share of those identifying with Platform Ops roles increases with company size. This is not surprising, since larger organizations typically have multiple Development teams. In turn, this drives the requirement for Platform Ops roles, as the need to maintain curated development platforms becomes more pressing.

In terms of SecOps roles, there is around a 7 percentage point (p.p) drop when comparing large enterprises to medium-sized and small businesses. Diving deeper, we find that 44% of employees at large enterprises have nothing to do with security, compared to 29% and 27% for those working at medium-sized and small businesses, respectively. This doesn't mean that security isn't important at large enterprises, but rather that responsibility for it is distributed: 40% of large enterprise employees collaborate with dedicated security teams, compared to 35% and 25% of medium-sized and small businesses.

The share of those in Platform Ops roles rises with company size

% of professional respondents self-identifying with each role category by company size (n=1,570)

Business size

Role	Small business (2-50 employees)	Medium-sized business (51-1,000 employees)	Large enterprise (More than 1,000 employees)
Development	64%	61%	63%
Platform Operations	36%	40%	42%
Infrastructure & Automation	42%	32%	23%
Leadership	48%	36%	29%
SecOps	20%	18%	12%
Data Science	16%	10%	11%

- 🟦 <5pp below the average of other business sizes
- 🟦 2.5 – 5pp below the average of other business sizes
- ⬜ ±2.5pp around the average of other business sizes
- ⬛ 2.5 – 5pp above the average of other business sizes
- 🟩 >5pp above the average of other business sizes

Use Cases

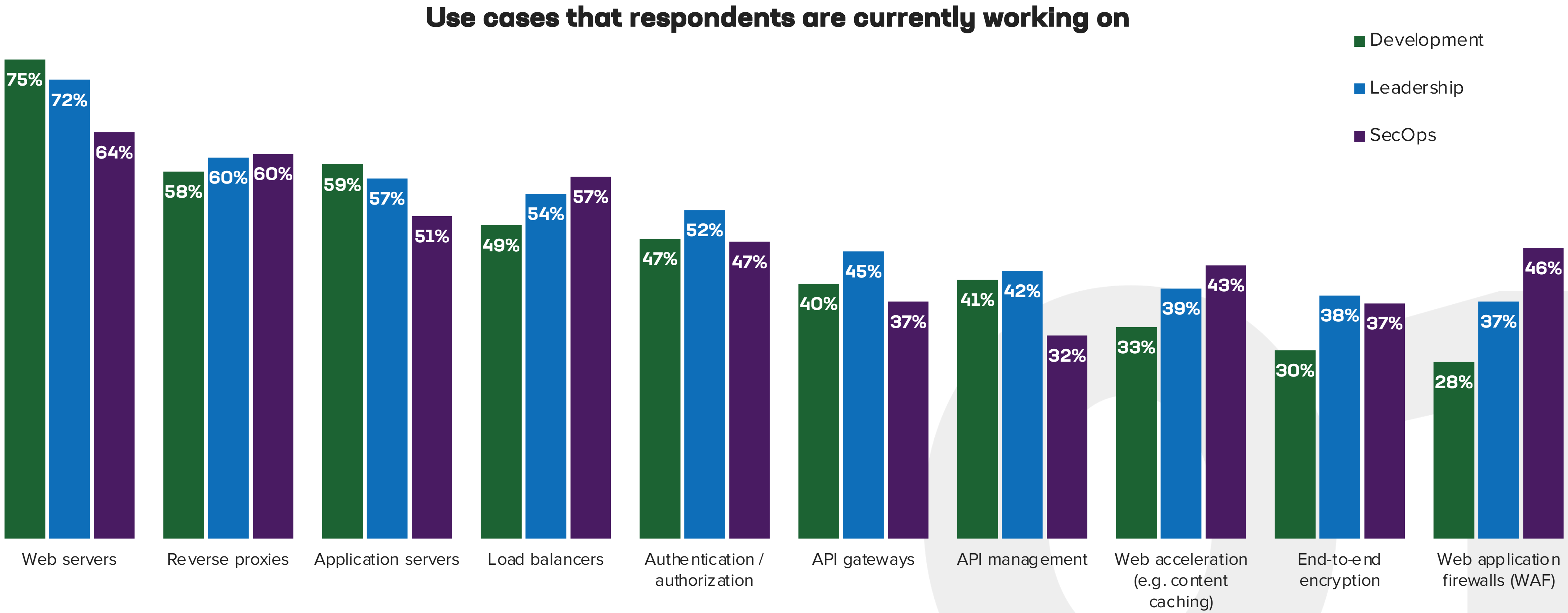
Moving on to the app and API delivery use cases that respondents are working on, we find that nearly 50% are currently using web servers, 36% reverse proxies, and 34% load balancers – not surprising given that NGINX Open Source is best known for those three use cases. However, the use cases change depending on respondents' roles, with particular highlights being the differences between those in Development, Leadership, and SecOps roles. For a full table of the use cases against roles, see the appendix.

As a general overview, those in Development roles are proportionally the most likely to be working on only two of the use cases we asked about. Here, it may be that these roles are more narrowly focused: in fact, we see that people in Development roles are less likely to work on four or more use cases (67%) compared to those in Leadership (72%) or SecOps roles (71%).

Meanwhile, the share of SecOps people currently working on load balancers is worth noting (57%), being higher than either those in Leadership (54%) or Development (49%) roles. While load balancing may be more associated with roles that are concerned with a broader purview, its importance in security operations is clear. After all, optimizing processing capability and distribution of security tools allows more traffic to be inspected more efficiently.

Respondents in SecOps roles are more likely to be working on more use cases than other roles

% of respondents (n=1,503)



Use cases that respondents are currently working on

Challenges in App and API Delivery

Now that we know what the NGINX community is working on, let's look at the top 10 challenges they face in their projects. As an overview, fewer than 1 in 5 respondents report that they have no challenges. For the other 83%, no one challenge dominates: the difference between the most and least “popular” challenges (lack of technical skills and real-time response to security violations) is only 7 p.p, demonstrating that the challenges faced are as varied as the respondents themselves.

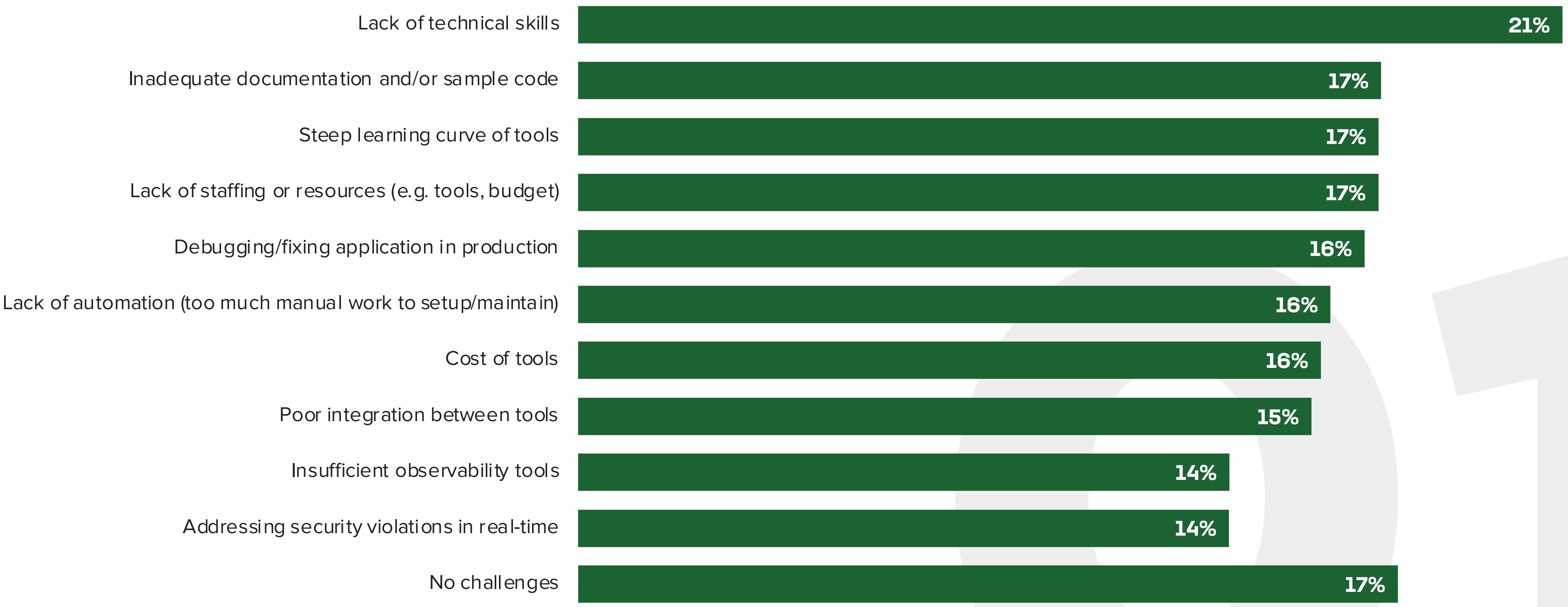
The largest issue respondents face is a lack of technical skills, with 21% reporting so. Of these respondents, 27% also reported the steep learning curve of tools being a major challenge, and 25% the lack of staffing or resources. Both of these challenges were mentioned (in tandem with lack of technical skills) at higher rates than the remaining options.

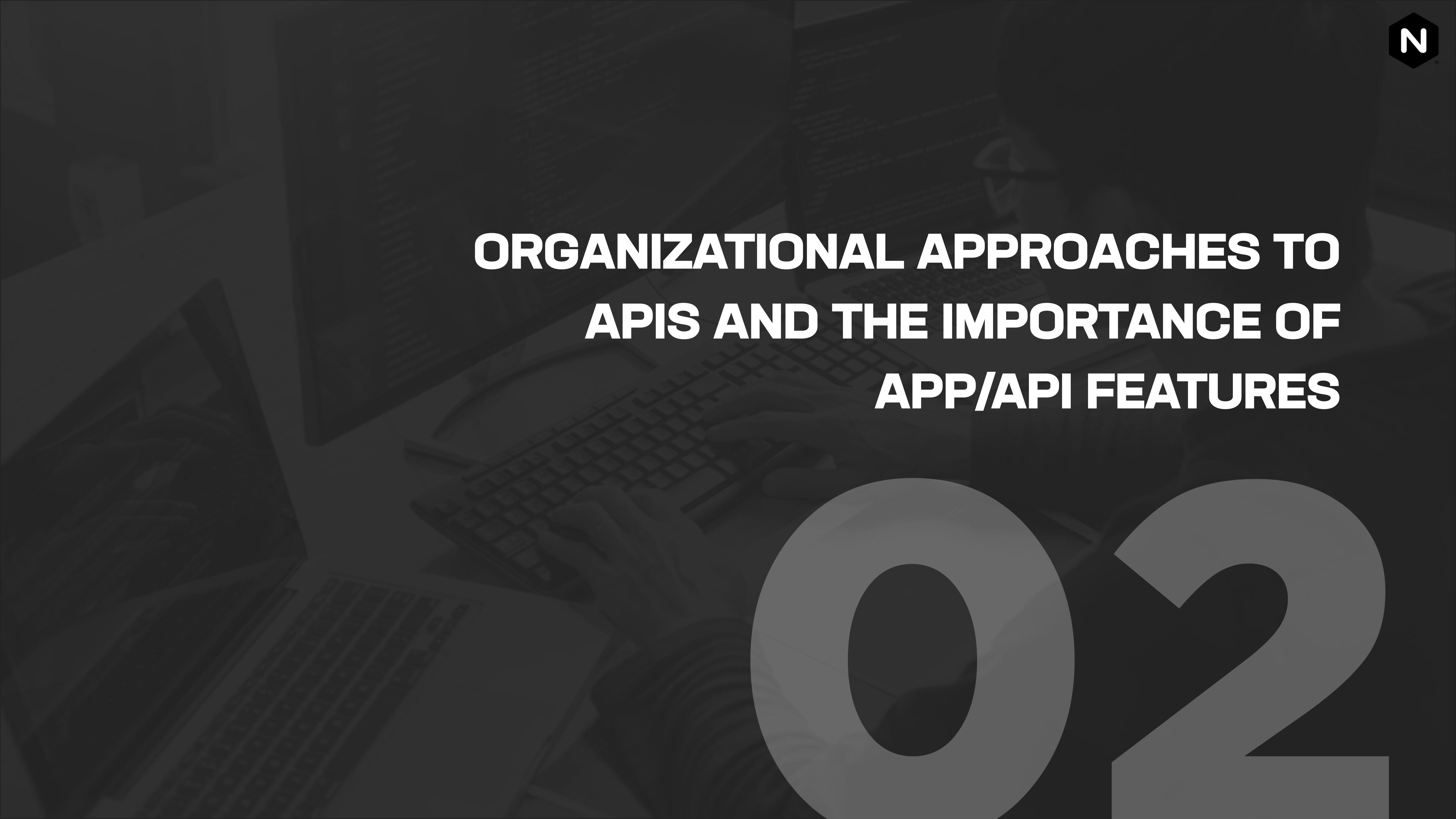
In short, the most frequent challenges faced are likely interrelated: a lack of technical skills can affect how steep the learning curve can seem, and a lack of staff or resources compounds these issues as there are fewer options for tackling the challenge.

Lack of technical skills is the top challenge of those working with API and app delivery projects by some margin

% of respondents (n=2,306)

Top 10 challenges with API and app delivery projects





ORGANIZATIONAL APPROACHES TO APIS AND THE IMPORTANCE OF APP/API FEATURES

02

Organizational Approach to APIs

In this chapter we focus on app and API delivery. First, we examine the degree to which organizations are adopting four key API-first practices: leveraging APIs as sources of revenue, designing the API first when building services, aligning APIs to their overall digital strategy, and designing APIs to be reusable. This analysis is broken down by organization size, where we find that larger organizations are more likely to be API-first. Then, we move on to which features or capabilities the NGINX community rate as important under the umbrellas of security, scalability, and observability.

We asked professional respondents about their organization's attitudes towards the four API-first practices. Nearly 80% tell us that their organizations follow the practices at least some of the time. Here we focus on those respondents who answered 'Always' for any of these approaches shown on the graph below, to see how approaches to APIs change for businesses who take them seriously.

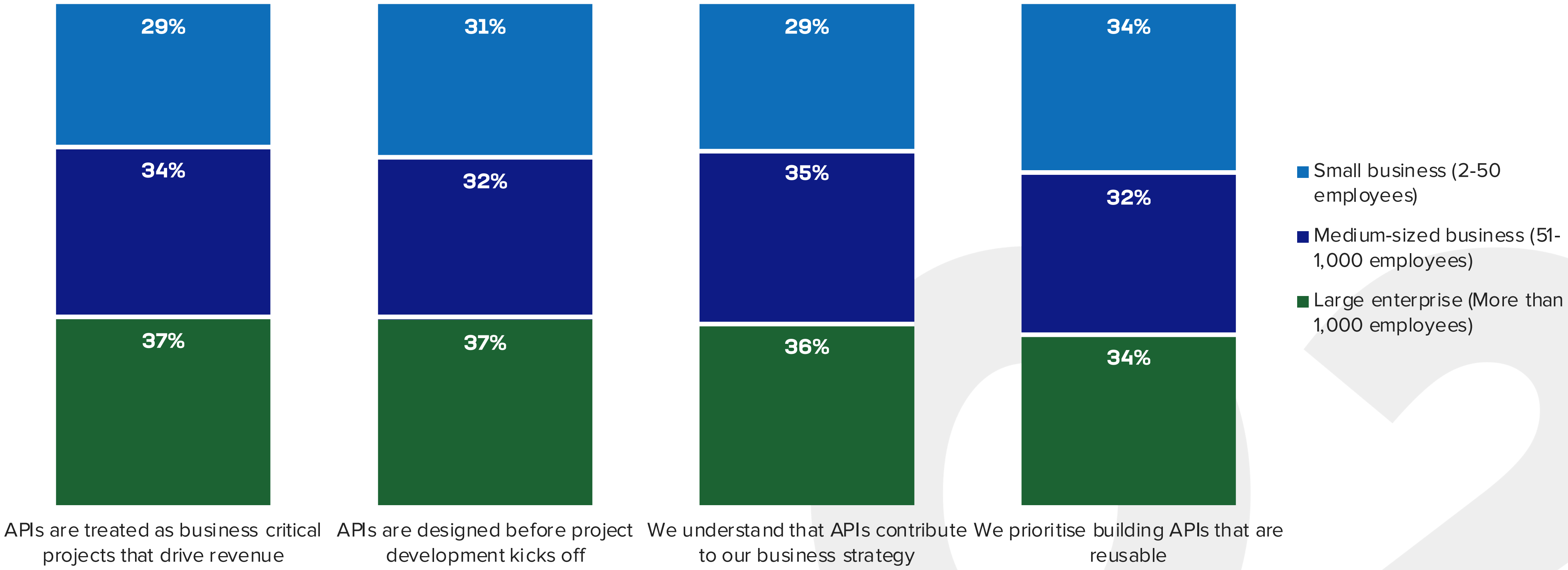
Respondents who work for large enterprises are the most likely to report that their companies always follow each of the practices – it's fair to assume that this is driven by the greater resources and capabilities of larger-sized businesses, which are more likely to use a greater variety of APIs.

This picture changes when we consider reusable APIs; here the share of respondents working for small businesses adopting this approach is larger than at medium-sized businesses. Again, this likely falls to the nature of these organizations – smaller businesses are more likely to need to keep costs low (in terms of both expenditure and workload), so careful planning and reusing of resources is a natural move.

Reusable APIs are a key consideration for smaller businesses

% of professional respondents who report that their company always follows each approach (n=1,390)

API approaches always followed within organizations



Importance of Security Features in App and API Delivery Projects

Having seen how respondents' organizations treat APIs, we can look at how important specific features are within their app and API delivery projects. In general, security features were rated as very important more often than any other features we asked about (see appendix for full chart): 45% rated at least one security feature as very important, in contrast to 34% of scalability features, and 40% of observability features.

Protection against the OWASP Top 10 and end-to-end encryption are the features that matter the most to respondents, with 53% rating one or the other as very important. At the other end of the spectrum, custom responses (obfuscation) is the feature least likely to be rated as very important (28%), and has the highest share of 'not important' ratings among security features (13%).

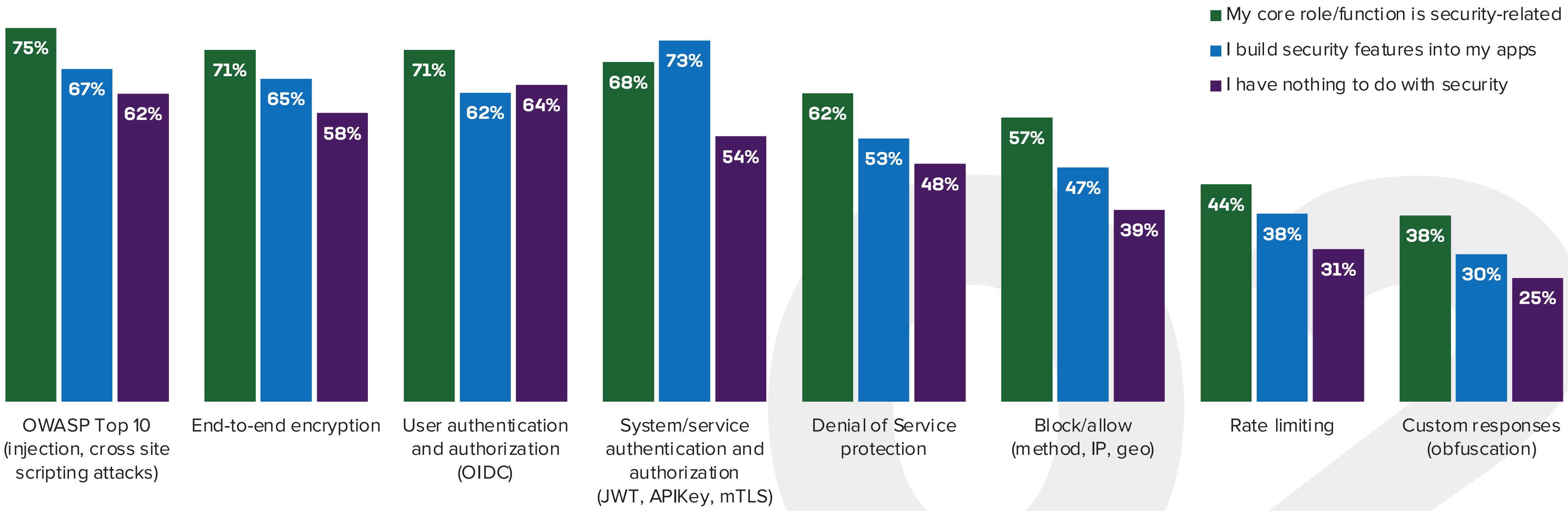
To see what those involved in security find important, we can look at who rated features as 'very important' by their level of security responsibility. Unsurprisingly, those whose core role is security-related are generally most likely to rate a feature as very important, while those with no security responsibility are generally the least likely.

However, there are cases that stand out. For example, a higher share of those with no security responsibility recognize that user authentication and authorization is very important, compared to those who build security features into their apps. Diving deeper, those who selected this option were 5 p.p more likely to have Platform Ops (40%) or Leadership (40%) roles than respondents as a whole. This makes sense: these respondents are more likely to be concerned about who accesses their projects than about security processes that are more technical.

Those working in SecOps are the most likely to rate security features in app and APIs delivery as very important

% of professional respondents working at businesses with at least 2 employees (I have nothing to do with security n=105 | I build security features into my apps n=345 | My core role/function is security-related n=184)

API security features rated as very important by level of security responsibility



Importance of Scalability Features in App and API Delivery Projects

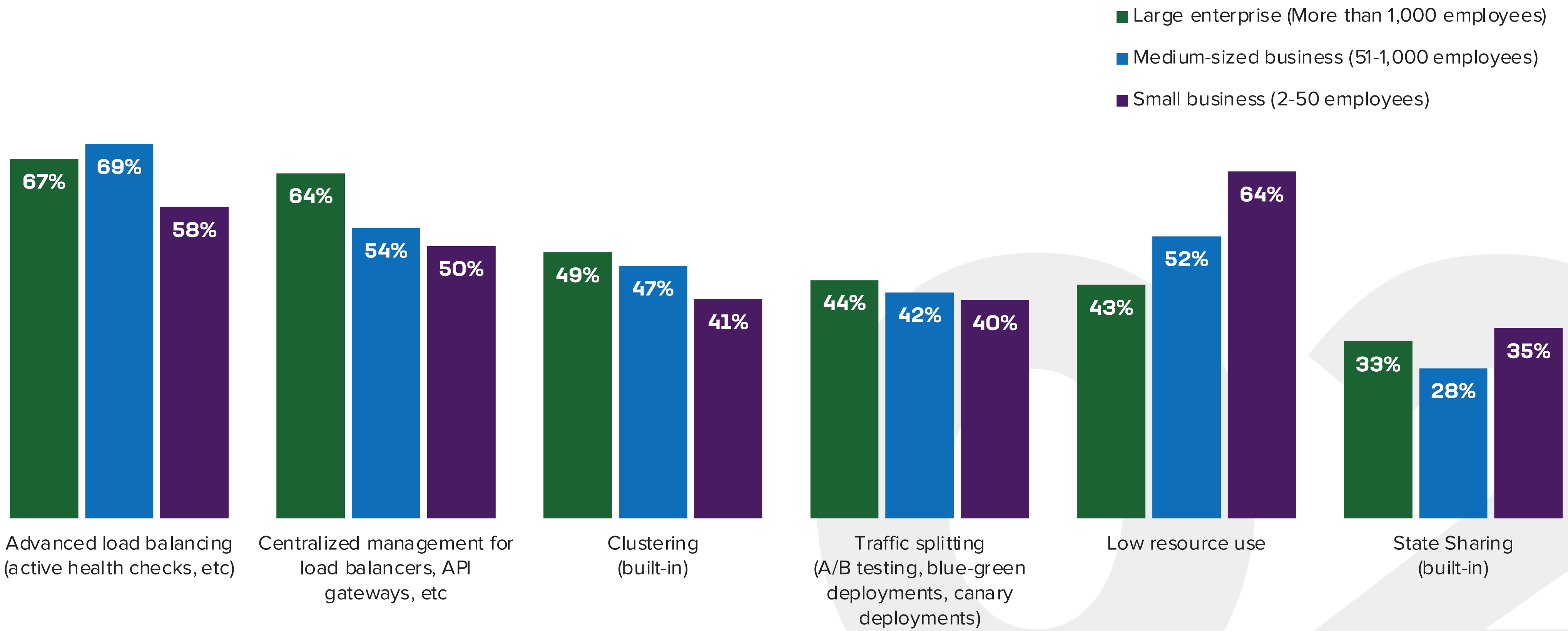
Moving on to the importance of scalability features, the top feature is advanced load balancing – 41% rate it as very important, the highest share across the set of features we asked about (see the appendix for a full chart). This makes sense – load balancing enables efficient and high-performance delivery by optimally distributing workload across relevant servers.

Echoing our analysis of API approaches by organization size, larger organizations are generally more likely to rate scalability features as very important. But again we see a reverse trend concerning low resource use. Employees at smaller size companies are much more likely to rate this feature as very important; in fact, it's the top scalability feature at smaller companies.

Low resource use is the top scalability feature for those working at smaller companies by some margin

% of professional respondents working at businesses with at least 2 employees (n=858)

Scalability features rated as very important by organization size



Importance of Observability Features in App and API Delivery Projects

Finally, let's take a look at which observability features the NGINX community rates as very important. Across all three categories we looked at (security, scalability, and observability), error/warning logging and performance monitoring are chosen as the two most important features in app and API delivery projects by the largest percentage of respondents. Only around 2% of those who rate these features say they are not important (see appendix for chart). The popularity of these features is likely due to their ubiquitous nature – who doesn't want to see how well a project is going, or where the bugs are?

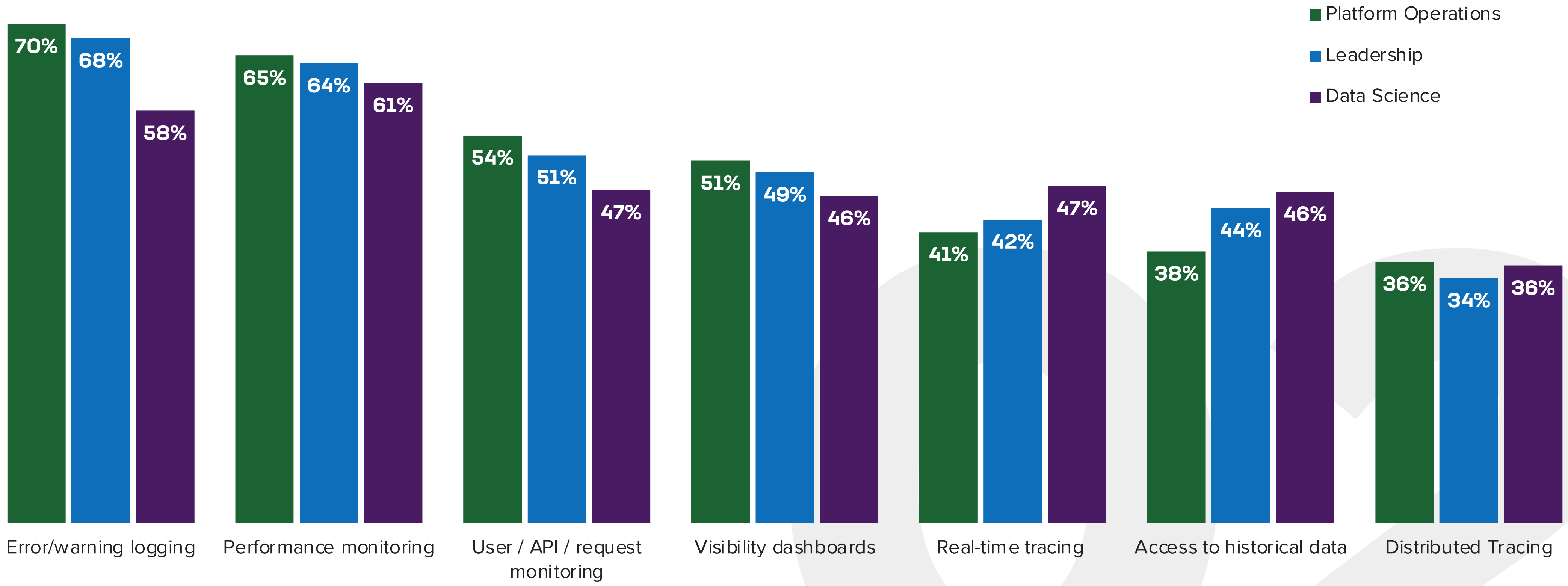
Looking at select roles where observability is key, we see that those in Platform Ops roles are the most likely to rate observability features as very important, which is to be expected. However, those in either Leadership or Data Science roles are more likely to rate real-time tracing and access to historical data as very important.

From a Data Science perspective, it makes sense that real-time tracing and historical data are rated highly – keeping on top of ever more complex pipelines and diagnostic data, both live and historical, allows these respondents to iterate and improve their processes. The view is similar from a leadership perspective – if mistakes/errors/bugs can't be tracked, how does a team learn from them?

Tracing and historical data are more important for those in Data Science roles compared to Platform Ops or Leadership roles

% of respondents in select roles (Platform Operations n=522 | Leadership (excl. security roles) n=494 | Data Science n=174)

Observability features rated as very important by role type



TECHNOLOGY CHOICES AND DEVELOPMENT ENVIRONMENTS

03

Workloads

In this third chapter, we look at the working styles of the NGINX community – workloads, where their code runs, their use of Kubernetes, and adoption of open source tools. In doing so, we aim to assess the maturity of their environments in terms of cloud-native approaches, container orchestration, and the preference (or even requirement) to use open source.

Note: We define a "workload" as a computing resource, of which there are a range of different categories from Kubernetes clusters to virtual machines.

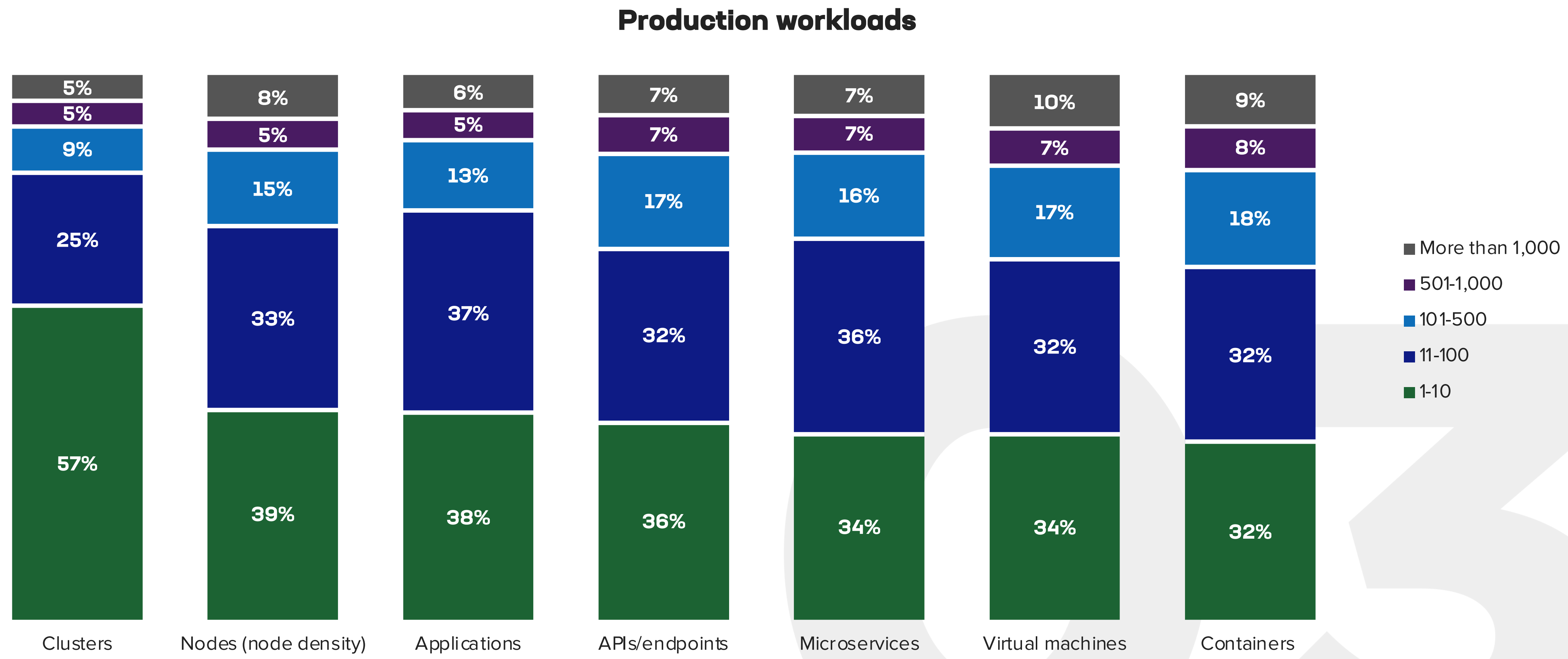
We asked respondents how many of the following types of workload they have running in production: Kubernetes clusters, nodes, applications, APIs/endpoints, microservices, virtual machines, and containers. The options for quantity were ranges: none, 1–10, 11–100, 101–500, 501–100, and1,000+. Here, we report the results for respondents who answered at least 1–10 for one or more workloads.

For all types of workloads except one, the percentage of respondents in each quantity range is basically the same. The exception is Kubernetes clusters, where the proportion of respondents in the 1–10 range is 13 percentage points higher than for the other workloads. As clusters are composed of sets of nodes, seeing a higher proportion compared to the other workloads (especially nodes) is unsurprising, and suggests ecosystems with appropriate architecture.

To simplify further analysis of the results, we grouped respondents into four categories based on the highest range they selected for **any** workload: low (10–100), medium (101–500), high (501–1000), and very high (over 1,000). On this basis, 20% of respondents have a low workload, 32% a medium workload, 21% a high workload, and 27% a very high workload.

Nearly double the share of respondents use 1-10 clusters compared to other workload types

% of respondents (n=1,280)



Kubernetes Adoption

When it comes to managing, scaling, and deploying containerized applications, Kubernetes has become the framework of choice for developers and businesses alike: 77% of respondents who use a container orchestration tool or service use a Kubernetes-based one, the same proportion as for Kubernetes users in general.

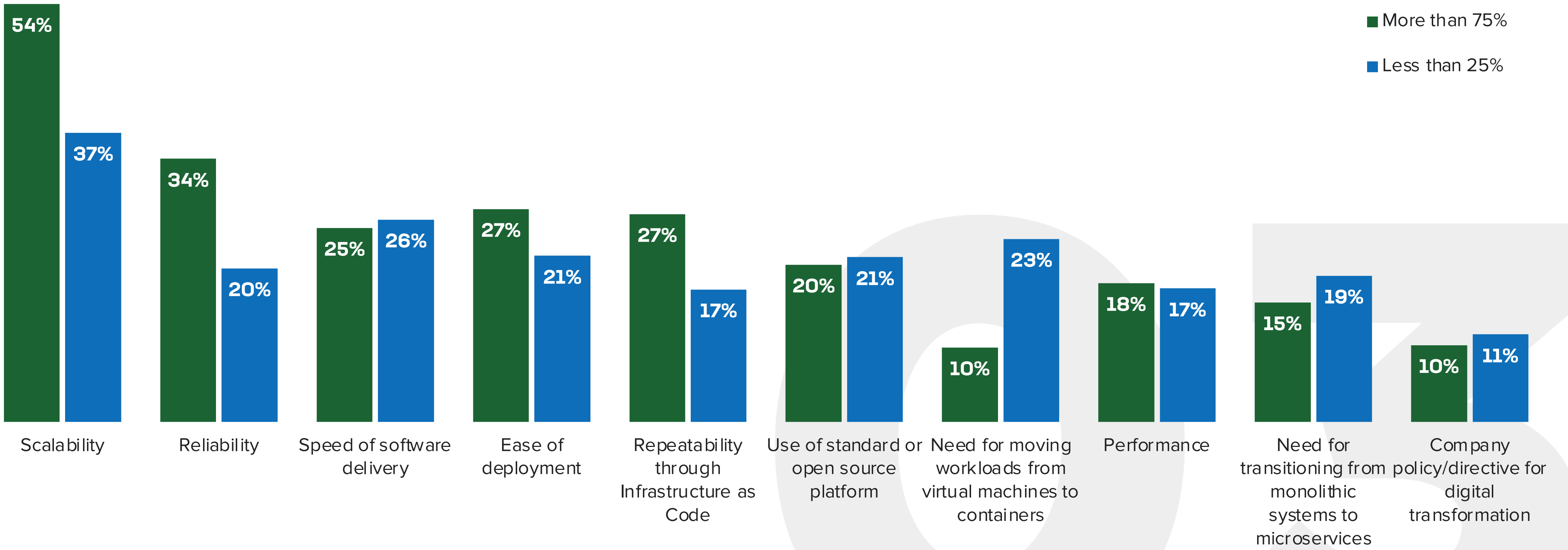
Looking at the drivers behind Kubernetes adoption in respondents' organizations, the number one answer by a large margin is scalability, with 42% of respondents selecting this option – 17 p.p more than the next answer, ease of deployment. However, this rate changes when considering the percentage of workloads which are deployed with Kubernetes.

For example, those who use Kubernetes to deploy more than three quarters of their workloads are 17 p.p more likely to rate scalability as the primary driver, when compared to those who use Kubernetes to deploy less than a quarter of their workloads (54% vs 37%). In this instance, respondents are walking the walk as well as talking the talk, in that their workload size drives the need for scalability: 46% of those who deploy at least three quarters of their work through Kubernetes have a very high workload, compared to 27% of those deploying less than a quarter.

Scalability is by far the most important driver when it comes to adopting Kubernetes

% of professional respondents who use Kubernetes-based container orchestration tools (n=436)

Motivation to adopt Kubernetes by % of workload deployed in Kubernetes



Where Do Respondents Deploy their Code?

We then asked respondents where they are running their apps or services. The top 3 answers are public cloud (37%), web client/front-end (35%), and on-premises servers (33%). Respondents could select more than one deployment environment, and use of multiple environments is common: of those who selected public cloud, 42% also selected web client/front end and 37% on-premises servers. While the industry once looked like it would go fully cloud-based, these findings align with what F5 hears from its customers: hybrid environments offer a balance of agility, cost savings, and security.

To explore regional differences in use of deployment environments, we compare how respondents deploy code in Western Europe & Israel, Middle East & Africa, and South Asia. In general, those in Western Europe & Israel are the most likely to deploy code in each environment we asked about – indicating a wide variety of ecosystems in the region. A smaller share of those in the Middle East & Africa or South Asia deploy in each environment, but here we will discuss specific cases where these regions ‘outperform’ the Western Europe & Israel Region.

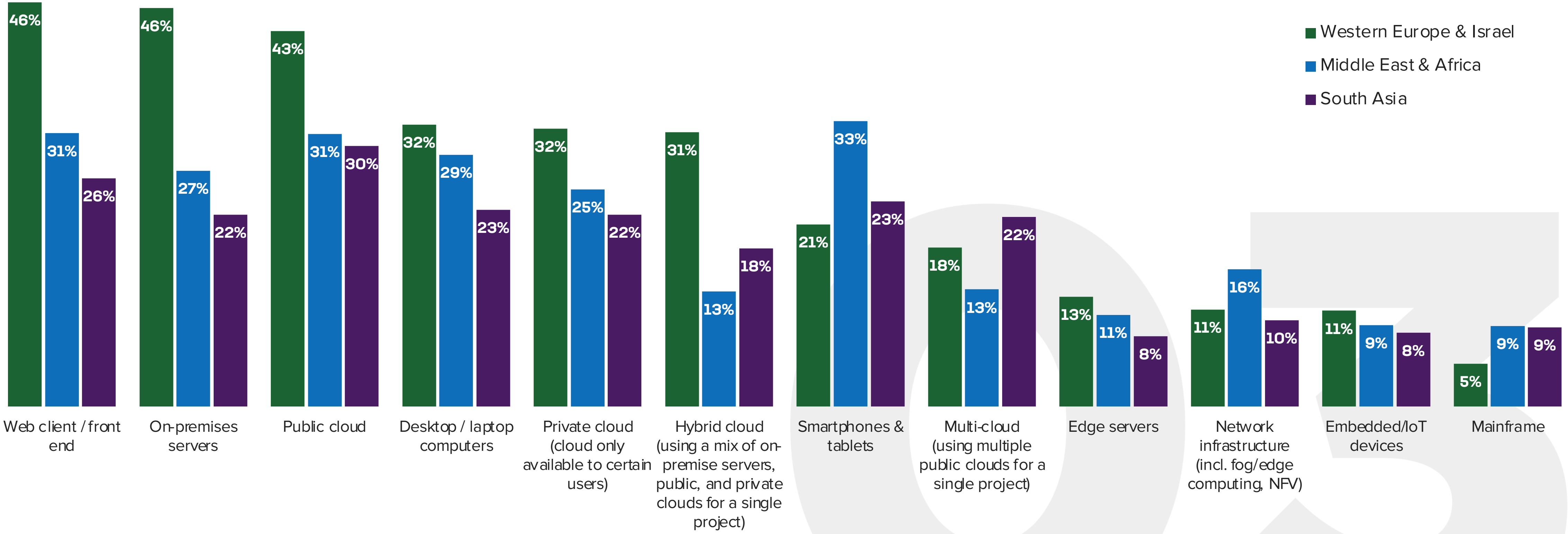
Focussing on cloud technologies, respondents based in Western Europe & Israel are generally more likely to deploy their code using each type of cloud environment we asked about, showcasing their approach and development towards a cloud-native environment. However, those in South Asia are most likely to deploy in a multi-cloud environment – potentially as an ongoing measure to tackle the transition from legacy systems to contemporary solutions, or perhaps in an effort to achieve the balance that can come with multi-cloud environments.

Respondents in the Middle East & Africa are at least 10 p.p more likely to deploy their code using smartphones & tablets than those in Western Europe & Israel or South Asia. The use of mobile devices for app and API projects is often a result of limited access to the right tooling, which in turn can drive smaller workloads – smartphones and tablets are hardly optimized for heavy work. Indeed, those in the Middle East & Africa region are the most likely to have a low workload (33%), compared to Western Europe & Israel (20%), and South Asia (14%).

A third of respondents in the Middle East & Africa region deploy their code on smartphones & tablets

% of respondents deploying their code in each environment by region (Western Europe & Israel n=403 | Middle East & Africa n=230 | South Asia n=310)

Top 10 deployment environments by select regions

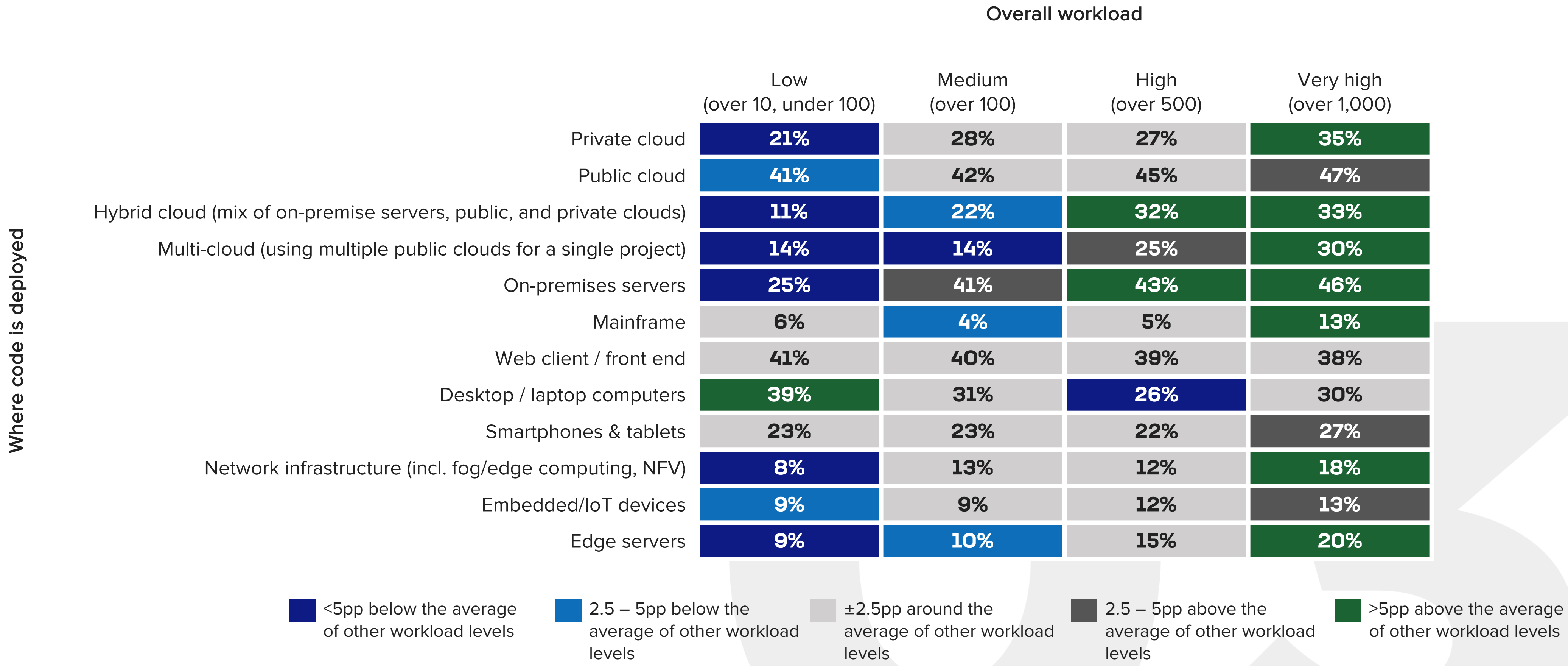


Speaking of overall workload, we see that the use of public cloud increases in parallel with overall workload size: low (41%), medium (42%), high (45%), and very high (47%). Interestingly, this doesn't necessarily indicate a move away from on-premises hardware, which also scales in parallel with overall workload size: low (25%), medium (41%), high (43%), and very high (46%).

Looking deeper, those with low workloads are most likely to work at small businesses (36%). This may be why a large percentage (39%) of low-workload respondents use desktop/laptop computers compared to those with higher workloads – such devices provide enough capacity for low workloads at a much lower price point than on-premises servers.

Dependency on cloud solutions increases with increasing overall workload

% of respondents with production workloads by code deployment environment (n=1,590)



Where code is deployed	Overall workload			
	Low (over 10, under 100)	Medium (over 100)	High (over 500)	Very high (over 1,000)
Private cloud	21%	28%	27%	35%
Public cloud	41%	42%	45%	47%
Hybrid cloud (mix of on-premise servers, public, and private clouds)	11%	22%	32%	33%
Multi-cloud (using multiple public clouds for a single project)	14%	14%	25%	30%
On-premises servers	25%	41%	43%	46%
Mainframe	6%	4%	5%	13%
Web client / front end	41%	40%	39%	38%
Desktop / laptop computers	39%	31%	26%	30%
Smartphones & tablets	23%	23%	22%	27%
Network infrastructure (incl. fog/edge computing, NFV)	8%	13%	12%	18%
Embedded/IoT devices	9%	9%	12%	13%
Edge servers	9%	10%	15%	20%

🟦 <5pp below the average of other workload levels

🟦 2.5 – 5pp below the average of other workload levels

⬜ ±2.5pp around the average of other workload levels

⬛ 2.5 – 5pp above the average of other workload levels

🟩 >5pp above the average of other workload levels

Open Source Tool Use

Finally, we asked the NGINX community about their organization's usage of and attitude towards open source software. The vast majority (80%) currently use it. Among those who don't, the reasons are various: 3% don't know how to make it work for their company, 4% don't trust it, and 3% have previously used it but since abandoned it. The remaining 10% have either never used it (4%), or are not sure/don't know (6%).

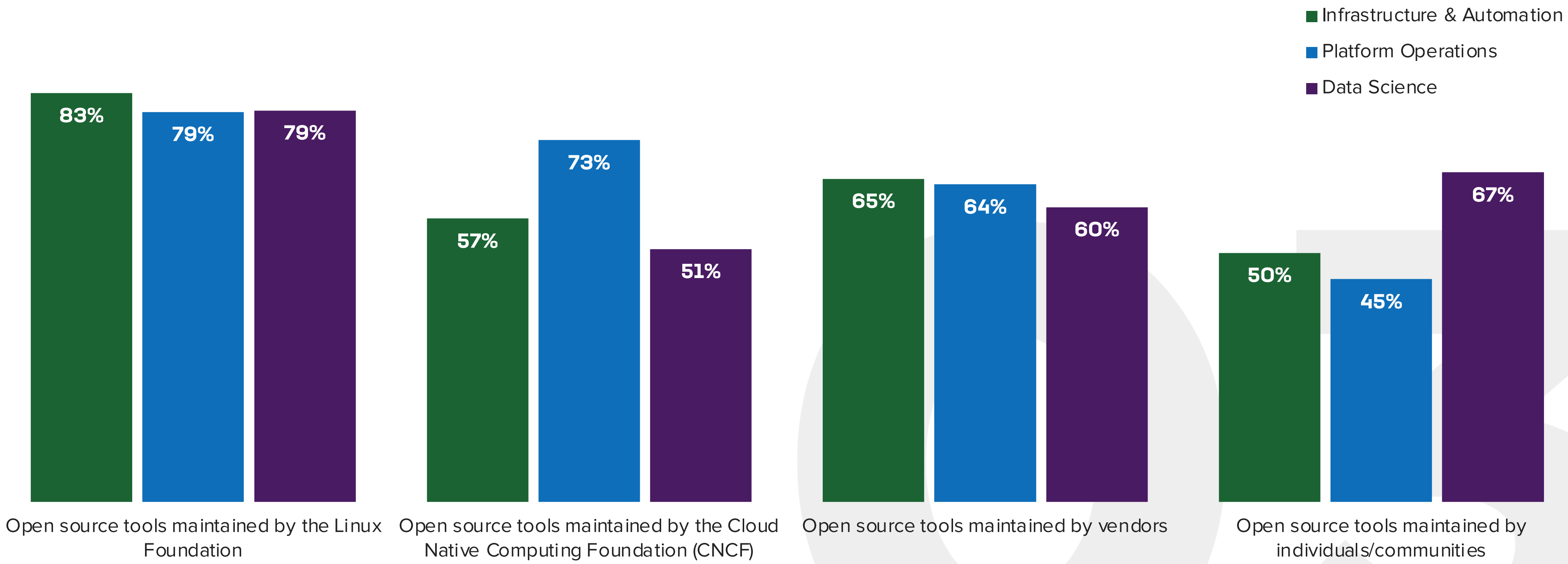
To look deeper into the use of open source, we asked the respondents who use open source tools 'whenever they can' where they get their tools from: the Linux foundation, vendors, the Cloud Native Computing Foundation (CNCF), or individuals/communities. Answers vary by select roles. While those in Platform Ops, Infrastructure & Automation, and Data Science get their open source tools from the Linux Foundation or vendors at a similar rate, there are standouts for the CNCF and tools maintained by individuals/communities.

For instance, respondents in Platform Ops roles are much more likely to get their tools from the CNCF. This makes sense, as the CNCF are principally more focussed on container technology, and, our data shows that those in Platform Ops are the most likely to use a container orchestration tool (80% doing so, see appendix for the full table). Meanwhile, those in Data Science are much more likely to get their tools from individuals/communities. At the broadest level, those in Data Science roles typically use community-managed tools like RStudio and Jupyter Notebook, which explains their higher proportion.

Those in Platform Ops or Data Science roles have strong preferences for their open source tools

% of respondents who use open source tools whenever they can by role (Platform Operations n=306 | Infrastructure & Automation n=234 | Data Science n=72)

Where do serious open source software users get their open source tools?



MANAGEMENT, SECURITY, AND MONITORING/OBSERVABILITY TOOL USAGE

04

Authentication and Authorization Tools

Having taken a broad overview of the NGINX community's use cases and working environments, we now look at the specific types of tools they use for security, management, and monitoring/observability.

From a broad perspective, 44% of respondents told us that they're currently working on authentication or authorization use cases, and, we've previously seen that respondents in Platform Ops and Leadership roles place particular importance on those features in their app and API delivery projects. So, which authentication and authorization tools does the community use? And are these same roles more likely to use said tools?

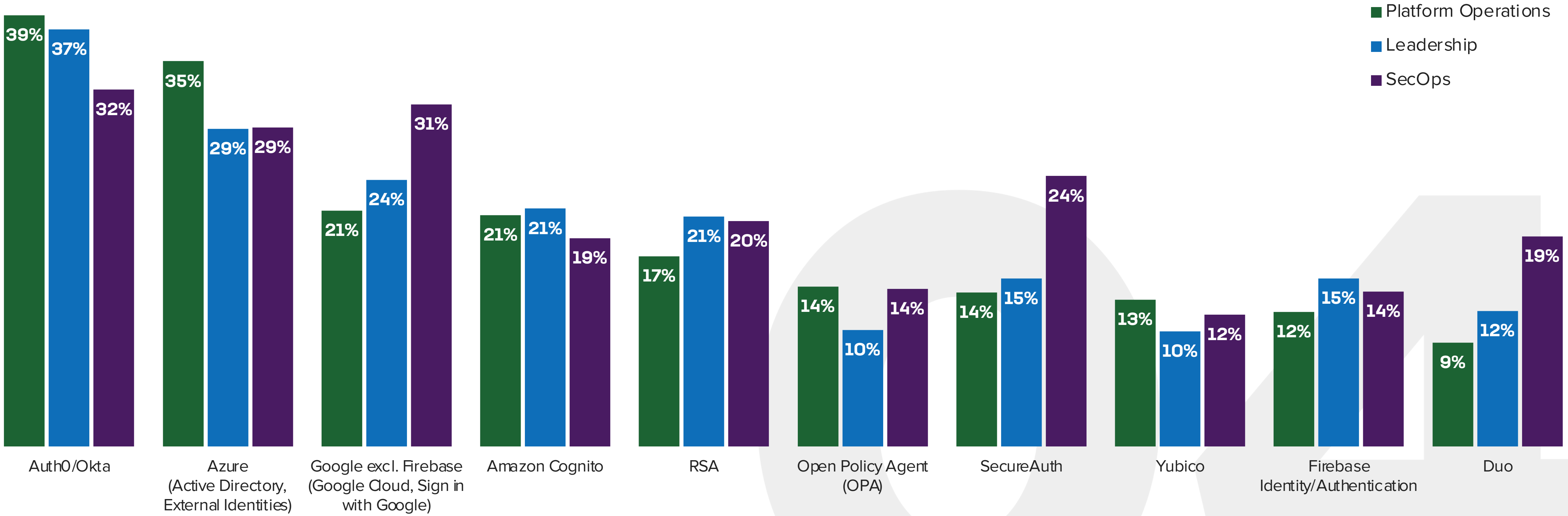
Broadly, and in sync with their rating of the importance of authentication and authorization, those in Platform Ops are the most likely to use 4 out of the top 10 tools (Auth0/Okta (39%), Azure (35%), Open Policy Agent (14%), and Yubico (13%). Those in Leadership roles are not far behind, being the most likely to use 3 out of the top 10: Amazon Cognito (31%), RSA (21%), and Firebase (15%).

Those in SecOps roles strongly favor 3 tools in particular: Google (excluding Firebase), SecureAuth, and Duo, typically selecting these tools at least 7 p.p more than the next role type. Looking into possible drivers for these choices, we found that these 3 tools typically provide 24/7 live support, whereas their competitors do not (with the exception of Auth0/Okta). If we presume that those in SecOps may be at the front line for emergency authentication/authorization issues, it makes sense that they value tools which can provide them with immediate support.

Respondents in SecOps roles showed strong preferences for specific authentication/authorization tools, likely linked to customer support

% of respondents currently working on authentication/authorization use cases (n=673)

Top 10 authentication/authorization tools used by select roles



Configuration Management Tools

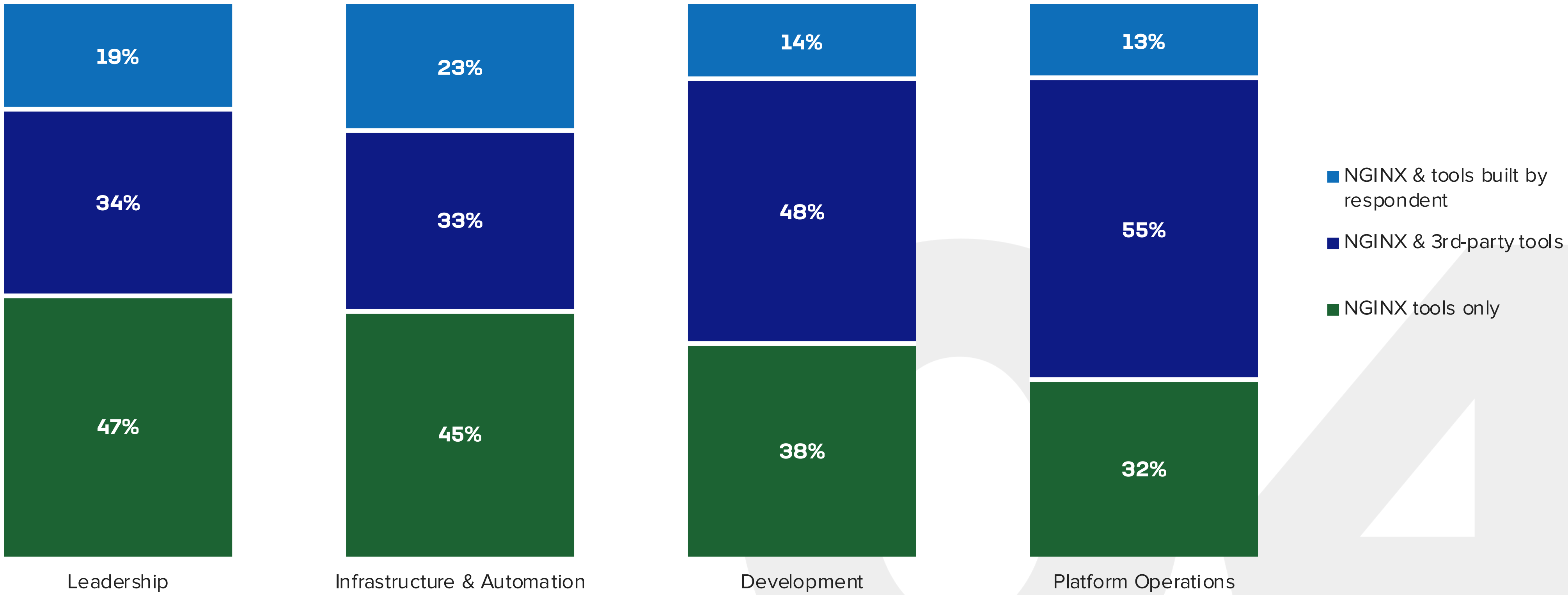
Next, let's look at which configuration management tools respondents use by role. Here we focus on respondents that use either NGINX tools only (such as NGINX Management Suite), NGINX tools in combination with third-party tools (like Terraform, Ansible), or NGINX with custom tools built by the respondent (in-house tools).

Within this group, those in Leadership roles are the most likely to use NGINX tools only. As these respondents need to take the broadest overview of projects, it makes sense that they prefer a single set of tools and a single vendor (i.e., “a single pane of glass”) rather than integrations that can imply more complexity and may require multiple vendor relationships. On the other hand, respondents who are more likely to be actively involved in multiple projects – that is, those in Platform Ops roles – are much more likely to use third-party tools in conjunction with NGINX tools based on experience (i.e., in-house subject matter expertise) or existing system integrations. Here, it’s likely that variety is key to keeping on track of a multitude of management issues.

Those in Leadership roles are more likely to depend solely on NGINX configuration management tools

% of respondents using NGINX configuration management tools (n=130)

Configuration management tools used by select roles



Monitoring and Observability

We also asked respondents which monitoring and observability tools they used in their app and API delivery projects: 27% use none or don't know, 51% use 1, 14% use 2, and 21% use 3 or more. The fact that over half use only one tool indicates that they have found a tool that satisfies their needs.

In terms of the most popular tools and their cross-usage, 30% of respondents said they use Grafana, of which 65% also use Prometheus, 43% use Elastic Stack, and 26% use Amazon CloudWatch. While our data can't show whether these respondents are using these tools in silo, it's telling that they are interoperable: Prometheus, Elastic Stack, and Amazon CloudWatch are all supported out-of-the-box by Grafana.

Grafana is the most popular monitoring tool, with the remaining tools regularly used in tandem with Grafana

% of respondents (n=2,306)

Monitoring/observability tool

Monitoring/ observability tool	Grafana	Prometheus	Elastic Stack	Amazon CloudWatch	Splunk
Grafana		80%	64%	45%	50%
Prometheus	65%		55%	39%	42%
Elastic Stack	43%	45%		37%	29%
Amazon CloudWatch	26%	28%	33%		30%
Splunk	23%	24%	20%	24%	

🟦 <5pp below the average of other tools

🟦 2.5 – 5pp below the average of other tools

⬜ ±2.5pp around the average of other tools

⬛ 2.5 – 5pp above the average of other tools

🟩 >5pp above the average of other tools

APPENDIX

Role categories

- Development
  - Application/software developer/engineer
  - Embedded software developer/engineer
  - DevOps or DevSecOps engineer/specialist
  - Test/QA developer or engineer
  - Architect (application/software)
- Platform Operations
  - Architect (cloud solution / cloud infrastructure)
  - Site reliability engineer (SRE)
  - Platform ops / engineer
  - Kubernetes cluster operator
- Infrastructure and Automation
  - Hardware engineer
  - Database administrator
  - System administrator
- Leadership (excluding security roles)
  - Tech/engineering team lead
  - CIO / CTO / IT manager
  - CEO/management
- SecOps
  - CISO / CSO / InfoSec manager
  - IT security engineer
- Data Science
  - Data/business analyst
  - Data scientist, machine learning developer or data engineer

Currently working on use case by role type

% of respondents (n=1,503)

Role Type

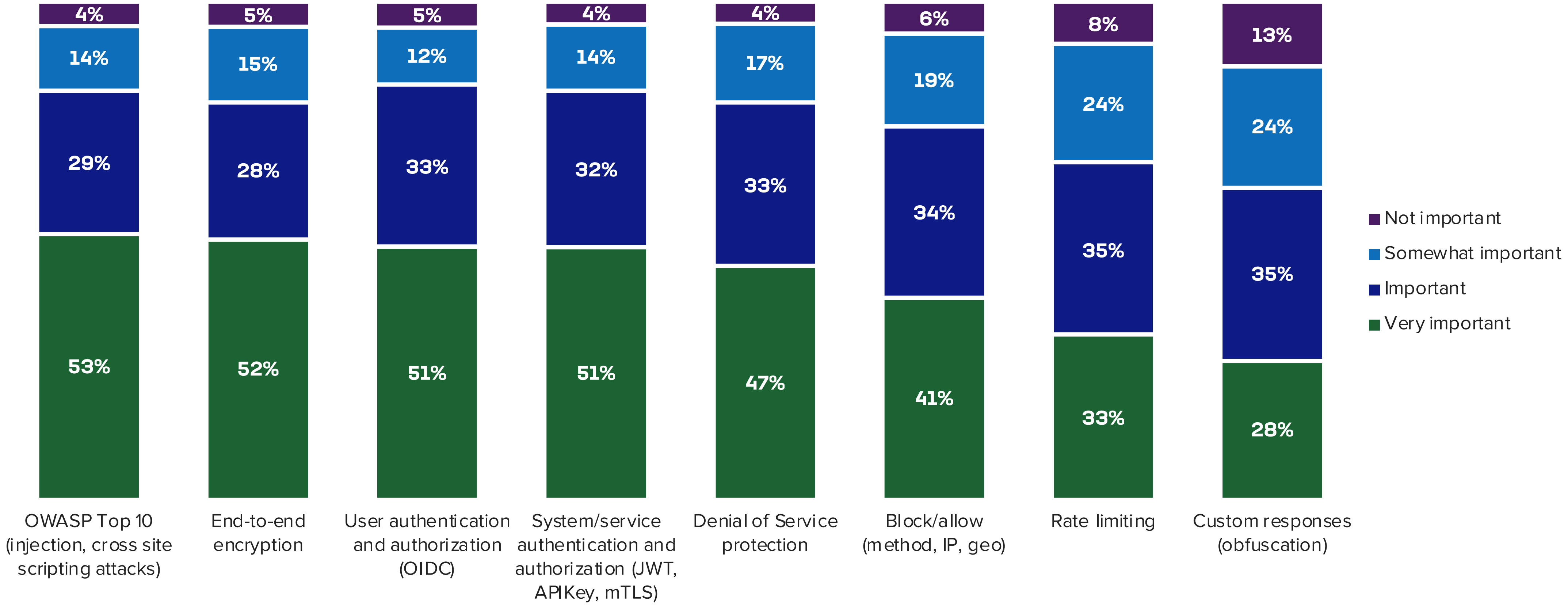
Currently working on	Development	Platform Operations	Infrastructure & Automation	Leadership	SecOps	Data Science
Web servers	75%	74%	75%	72%	64%	64%
Reverse proxies	58%	61%	56%	60%	60%	40%
Application servers	59%	57%	54%	57%	51%	47%
Load balancers	49%	61%	46%	54%	57%	37%
Authentication / authorization	47%	49%	40%	52%	47%	43%
API gateways	40%	47%	30%	45%	37%	40%
API management	41%	42%	29%	42%	32%	35%
Web acceleration	33%	38%	38%	39%	43%	35%
End-to-end encryption	30%	38%	32%	38%	37%	34%
Web application firewalls (WAF)	28%	38%	29%	37%	46%	28%

- 🟦 <5pp below the average of other role types
- 🔹 2.5 – 5pp below the average of other role types
- ⬜ ±2.5pp around the average of other role types
- ⬛ 2.5 – 5pp above the average of other role types
- 🟩 >5pp above the average of other role types

Importance of security features

% of respondents (n=1,793)

Importance of security features in app or API delivery projects

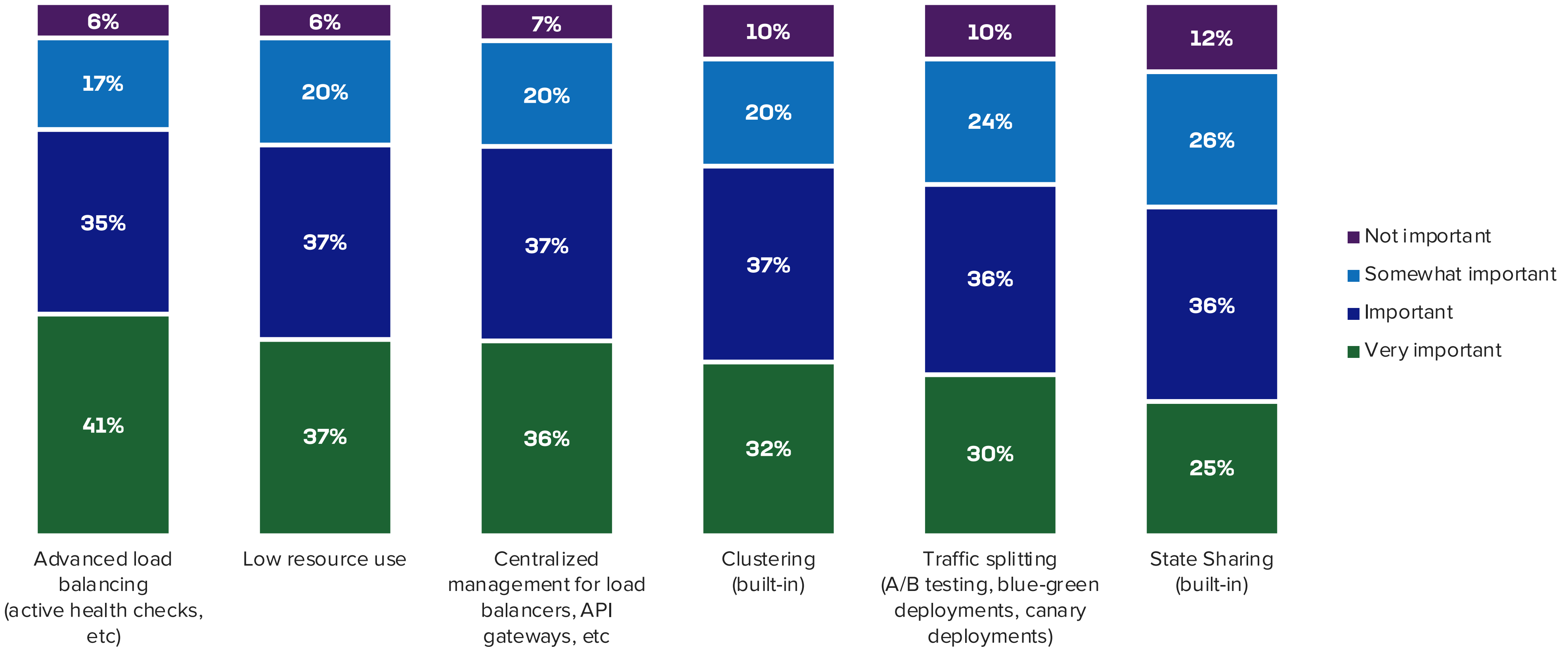


Appendix

Importance of scalability features

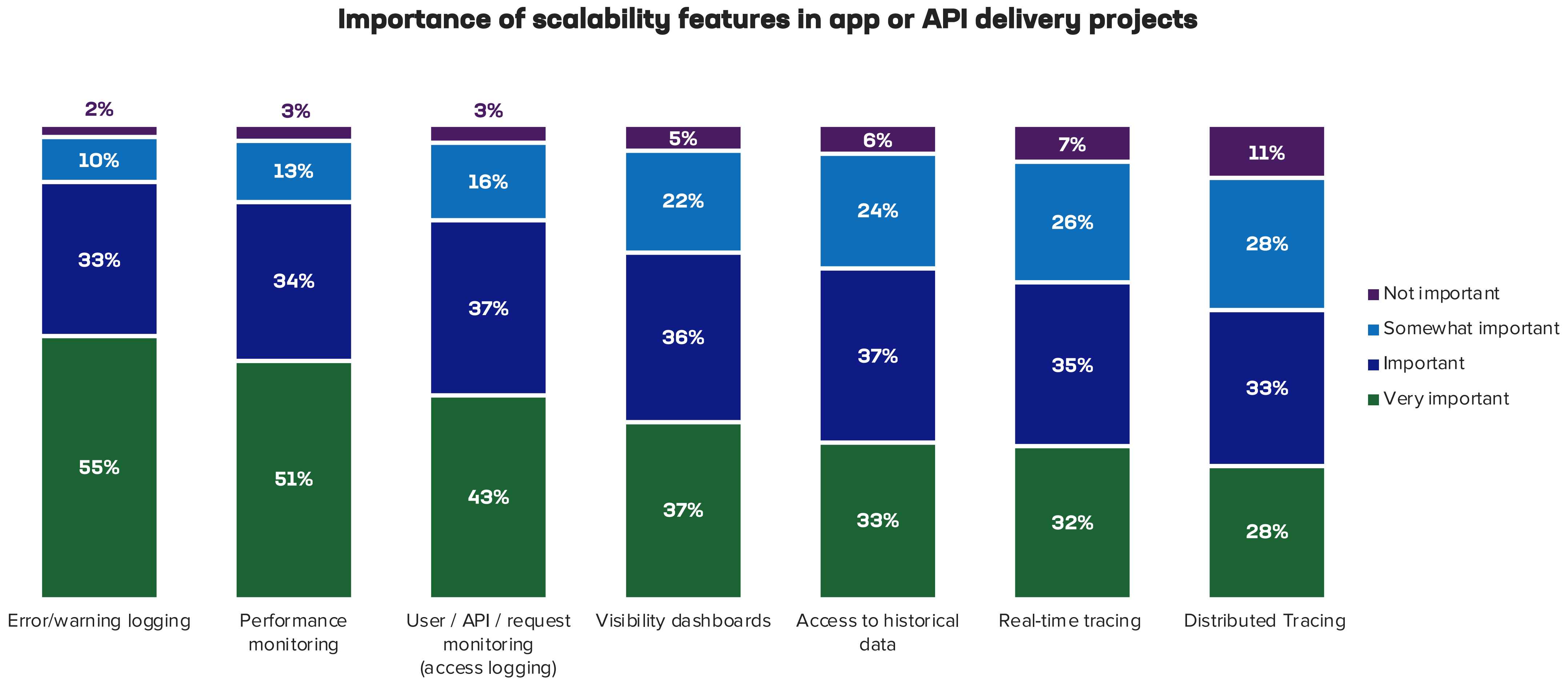
% of respondents (n=1,748)

Importance of scalability features in app or API delivery projects



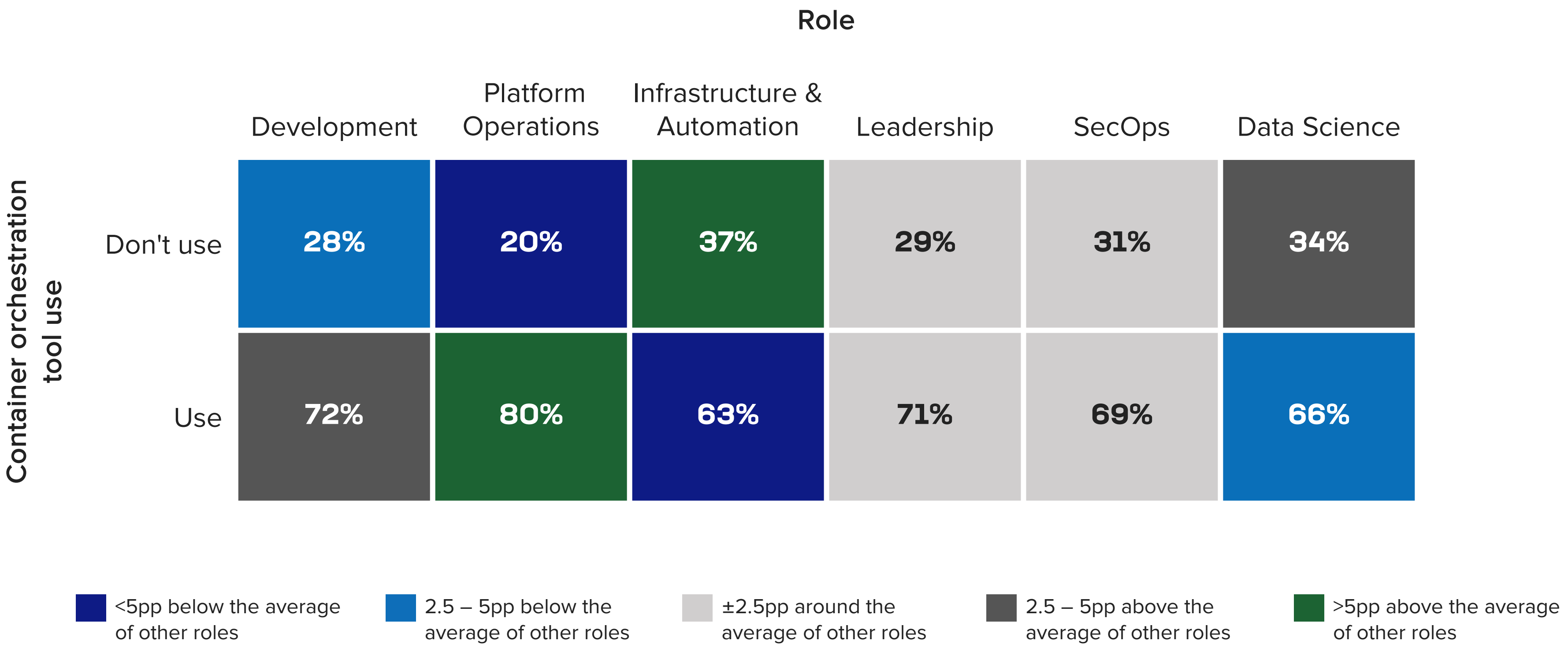
Importance of observability features

% of respondents (n=1,748)



Container orchestration tool use by role

% of respondents (n=2,092)



Container orchestration tool use	Development	Platform Operations	Infrastructure & Automation	Leadership	SecOps	Data Science
Don't use	28%	20%	37%	29%	31%	34%
Use	72%	80%	63%	71%	69%	66%

- ■ <5pp below the average of other roles
- 🟦 2.5 – 5pp below the average of other roles
- ⬜ ±2.5pp around the average of other roles
- ⬛ 2.5 – 5pp above the average of other roles
- 🟩 >5pp above the average of other roles

About SlashData

/DATA

We help the world understand developers

We survey 30,000+ developers annually – across web, mobile, IoT, cloud, Machine Learning, AR/VR, games and desktop - to help companies understand who developers are, what they buy and where they are going next.

WHO DEVELOPERS ARE

Developer population sizing
Developer segmentation

WHAT THEY BUY

Why developers are adopting competitor products – and how you can fix that

WHERE THEY ARE GOING

Emerging platforms – augmented & virtual reality, machine learning

NOVEMBER 2022

NGINX State of App and API Delivery Report

/DATA

NGINX
Part of F5