CNCF TECHNOLOGY RADAR

Observability, IaC, and Storage Tooling



/DATA

CLOUD NATIVE COMPUTING FOUNDATION

OCTOBER 2026

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 the Cloud Native Computing Foundation 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
SlashData © Copyright 2026 | Some rights reserved

ABOUT THE AUTHORS

liam.dodd@slashdata.co



Liam Bollmann - Dodd

Principal Research Consultant

Liam is a former experimental antimatter physicist, and he obtained a PhD in Physics while working at CERN. He is interested in the changing landscape of cloud development, cybersecurity, and the relationship between technological developments and their impact on society.

alvaro.ruiz@slashdata.co



Álvaro Ruiz Cubero

Research Manager

Álvaro is a market research analyst with a background in strategy and operations consulting. He holds a Master's in Business Management and believes in the power of data-driven decision-making. Álvaro is passionate about helping businesses tackle complex strategic business challenges and make strategic decisions that are backed by thorough research and analysis.

TABLE OF CONTENTS

KEY INSIGHTS 5

INTRODUCTION 6

OBSERVABILITY 9

INFRASTRUCTURE AS CODE 17

STORAGE TOOLS 25

PLATFORMS FOR HANDLING AI WORKFLOWS 33

METHODOLOGY 37

REPORT HIGHLIGHTS

- **Prometheus and OpenTelemetry lead observability, placed in Adopt.** Both tools scored strongly across maturity, usefulness, and recommendation despite never leading on any individual measure. →
- **Fluent Bit is the most recommended observability tool, despite only moderate ratings.** 99% of current and former users are likely or highly likely to recommend it, and 64% highly likely, against only 28% giving it 5 stars for maturity and 20% for usefulness. Fluent Bit appears to be treated as the default choice for log collection. →
- **OpenFeature and OpenTofu lead IaC, placed in Adopt.** Although neither project has graduated under CNCF's project maturity model, they resonated strongly with developers. →
- **OpenTofu combines the largest user base in IaC with the highest recommendation (96%).** It scores highly on both maturity and usefulness, despite being only a sandbox project in CNCF, suggesting developers are rating the maturity of the Terraform codebase it carries forward. →
- **For storage technologies, Carina stands alone in the Adopt criteria.** Carina leads on maturity and usefulness (33% 5-star ratings for both) with high levels of recommendation. →
- **The two most widely used storage tools draw very different recommendations.** Longhorn has a large share of users highly likely to recommend it (59%), despite modest maturity and usefulness ratings, while Rook's recommendation (94%) is driven mostly by likely ratings (63%). Longhorn's simplicity may make it an easy recommendation for any team, while Rook's reliance on Ceph means developers may recommend it with a caveat. →
- **Centralized platform teams commit to a single approach for AI workflows.** Only 5% of teams building a custom IDP, and 18% of teams curating third-party tools, take a hybrid approach, against 43% of organizations with multiple collaborating teams and 54% where individual teams manage their own tooling. →
- **Whether a platform is built or assembled shapes which approach is chosen.** Teams building a custom IDP are the most likely to build a separate dedicated AI platform (38%), while teams curating third-party tools are the most likely to rely on AI vendor platforms (25%). →

INTRODUCTION

01

1. Introduction

In Q3 2026, more than 300 professional developers using technologies associated with cloud native development were asked about their experience and opinions with regard to platform engineering tools for observability, Infrastructure as Code, and databases. The technologies shown to developers were selected by CNCF and CNCF's End User Community for relevance and importance. The developers surveyed originate from around the world and have a large range of specialties and areas of focus. A more granular breakdown of the respondents is included in the Methodology section.

For the products or tools they were familiar with, those they were either currently or previously using, they rated them on their usefulness and maturity and indicated how likely they were to recommend that technology to other developers. Within the context of this report, usefulness was defined as how well a given technology meets project requirements, and maturity was related to its stability and reliability. The recommendation scale was converted into a net promoter score (NPS) for use during the analysis.

1. Introduction

Based on the usage, usefulness, and maturity ratings and how likely respondents are to recommend a given technology, we categorized the technologies into three groups: adopt, trial, assess. 'Adopt' technologies are considered reliable choices by developers and applicable for most use cases. 'Trial' technologies are worth exploring to see if they meet your specific needs, and 'Assess' technologies require careful assessment for their relevance, applicability, and performance for specific project needs. This research provides insights into which platform engineering tools are gaining traction among professional developers and helps identify emerging patterns in technology adoption across the cloud native landscape.

Note: These radar positions do not necessarily correlate with the CNCF maturity model (Sandbox, Incubating, and Graduated), which corresponds to the Innovators, Early Adopters, and Early Majority tiers from Geoffrey A. Moore's *Crossing the Chasm: Marketing and Selling High-Tech Products to Mainstream Customers.*

Sandbox: Sandbox projects are in their earliest stages, meant for experimentation and foundational growth. They are newer technologies that represent initial concepts and technologies with significant room for evolution.

Incubating: Projects that have a solidified technical vision and a growing contributor base but are still maturing in terms of community adoption, stability, and governance.

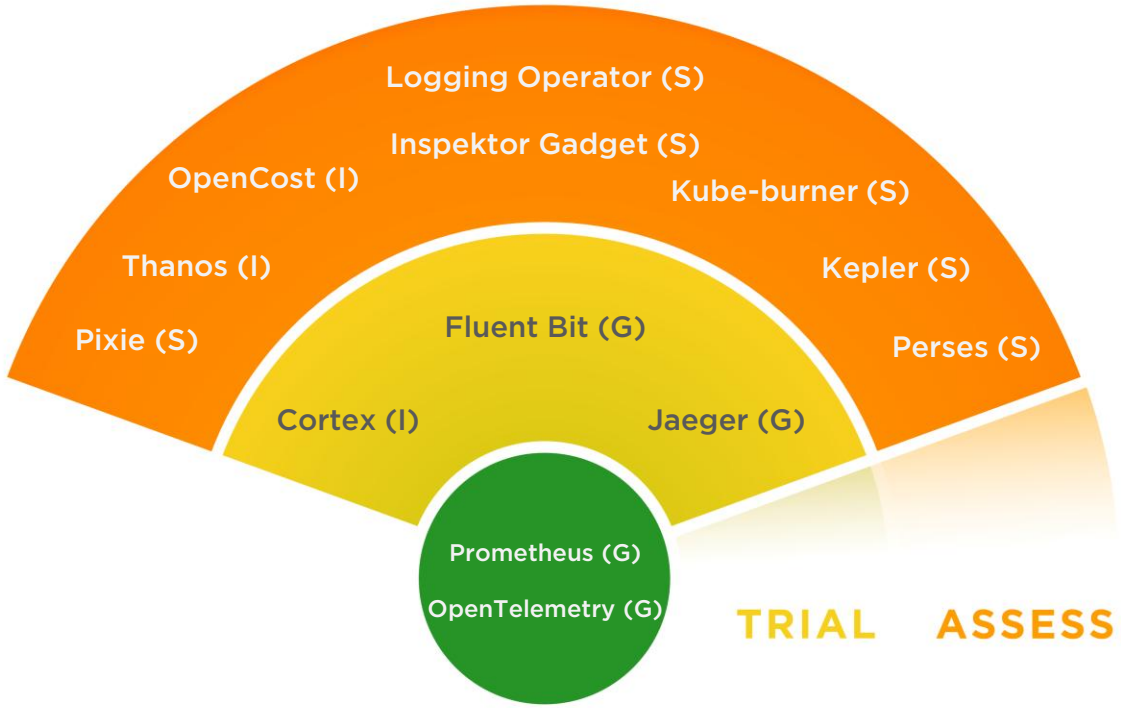
Graduated: Graduated projects are widely adopted and reliable. They have established a diverse community base supported by mature technical policies and governance.

OBSERVABILITY

02

2. Observability

For the observability technologies, we find Prometheus and OpenTelemetry were categorized into the ‘adopt’ position of the technology radar. Receiving slightly less positive results were Cortex, Fluent Bit, and Jaeger, which ended up in the ‘trial’ position of the technology radar.



Observability ADOPT

G Graduated I Incubating S Sandbox

Professional developers working with observability technologies (n=239)

Based on developer perceptions: 'adopt' technologies are considered reliable choices for most use cases, 'trial' technologies are worth exploring to see if they meet your specific needs, and 'assess' technologies require careful evaluation before committing

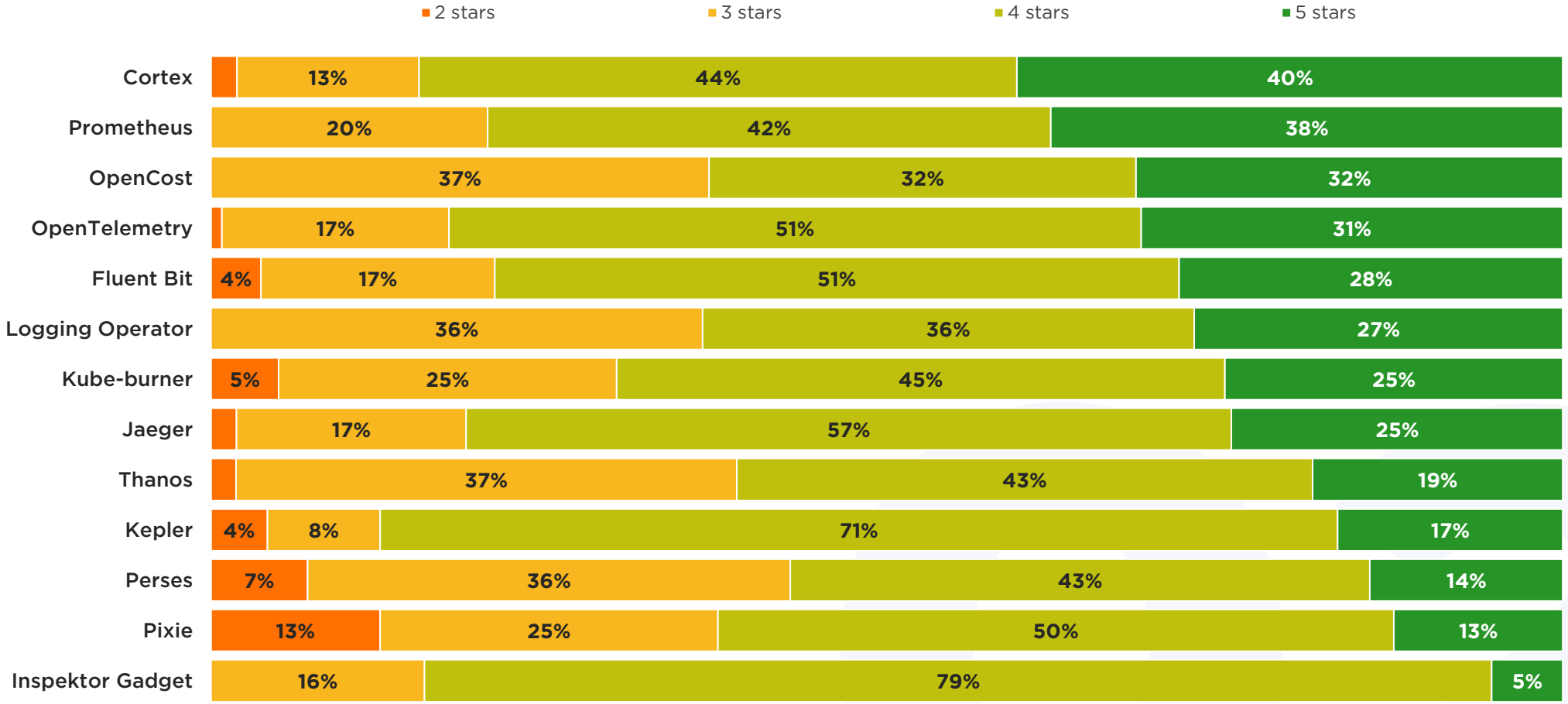
Maturity

On maturity ratings, Cortex received the greatest proportion of 5-star ratings (40%), ahead of Prometheus (38%) and OpenCost (32%). Cortex also received 44% of 4-star ratings, pushing it to have an overall high perception of maturity (84%). While Prometheus is close behind (80%), it also has a much larger pool of developers either using it or familiar with it within our survey, indicating that it is achieving high levels of satisfaction across a broader user group.

Inspektor Gadget (84%) and Kepler (88%) are among the leaders on combined 4- and 5-star ratings but are at the bottom of the group when isolated to just 5-star ratings, 5% and 17% respectively. For Sandbox tools with a smaller current audience, this is a positive reflection of their progress. Users are not seeing mediocre reliability and stability frequently, but the technologies are not yet reaching the highest levels of developer expectations.

Maturity Ratings of Observability Technologies

% of developers currently or previously using the technology (n=239)



Question wording: How would you rate the following observability tools with respect to these aspects? (Maturity)

Usefulness

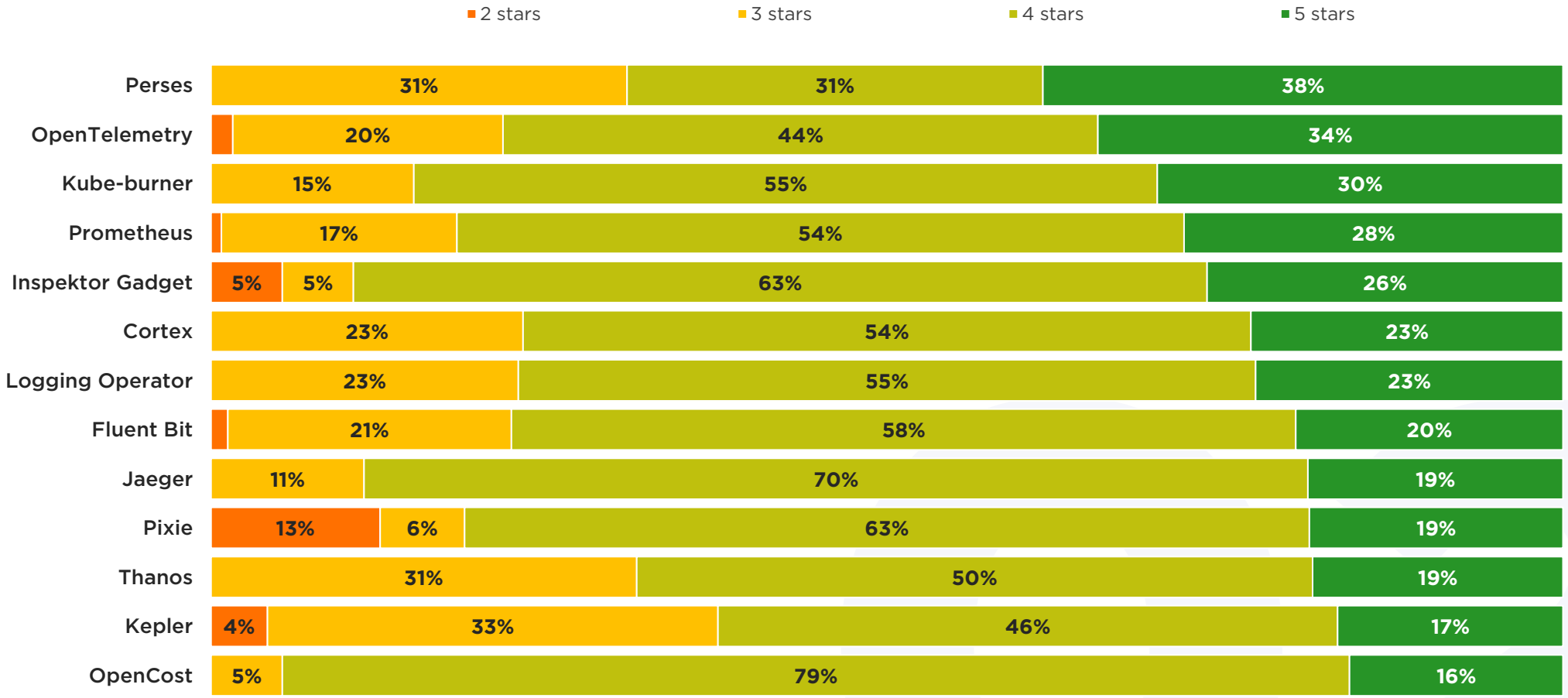
While Perses ranked at the bottom of maturity, it has the largest proportion of 5-star ratings for usefulness (38%), as well as a reasonable proportion of 4-star ratings (31%). This suggests that for a slice of developers, Perses' utility is incredibly high. This may be because of its ability as a visualizer for other observability tooling, being highly valuable to these developers even if the maturity of the tool is lagging.

OpenTelemetry has the second-highest rate of 5-star usefulness ratings (34%) but also has a much larger developer audience. OpenTelemetry is therefore succeeding at providing high levels of utility to a greater range of users, reflected in the scope of features and capabilities of the technology. Additionally, the popularity of this technology can also explain the percentage of negative ratings: with a larger group of developers recommending the tool (see Recommendations), the greater the chance of developers who do not find it useful being exposed to it.

OpenCost received the fewest 5-star ratings (16%) but the largest portion of combined 4- and 5-star ratings (95%). Cost control and measurement is a perennial issue among developers, and OpenCost acting as a cloud native solution for this is likely leading developers to consider it useful, but as a incubating project, it is likely missing features that would elevate it to 5 stars.

Usefulness Ratings of Observability Technologies

% of developers currently or previously using the technology (n=239)



Question wording: How would you rate the following observability tools with respect to these aspects? (Usefulness)

Recommendations

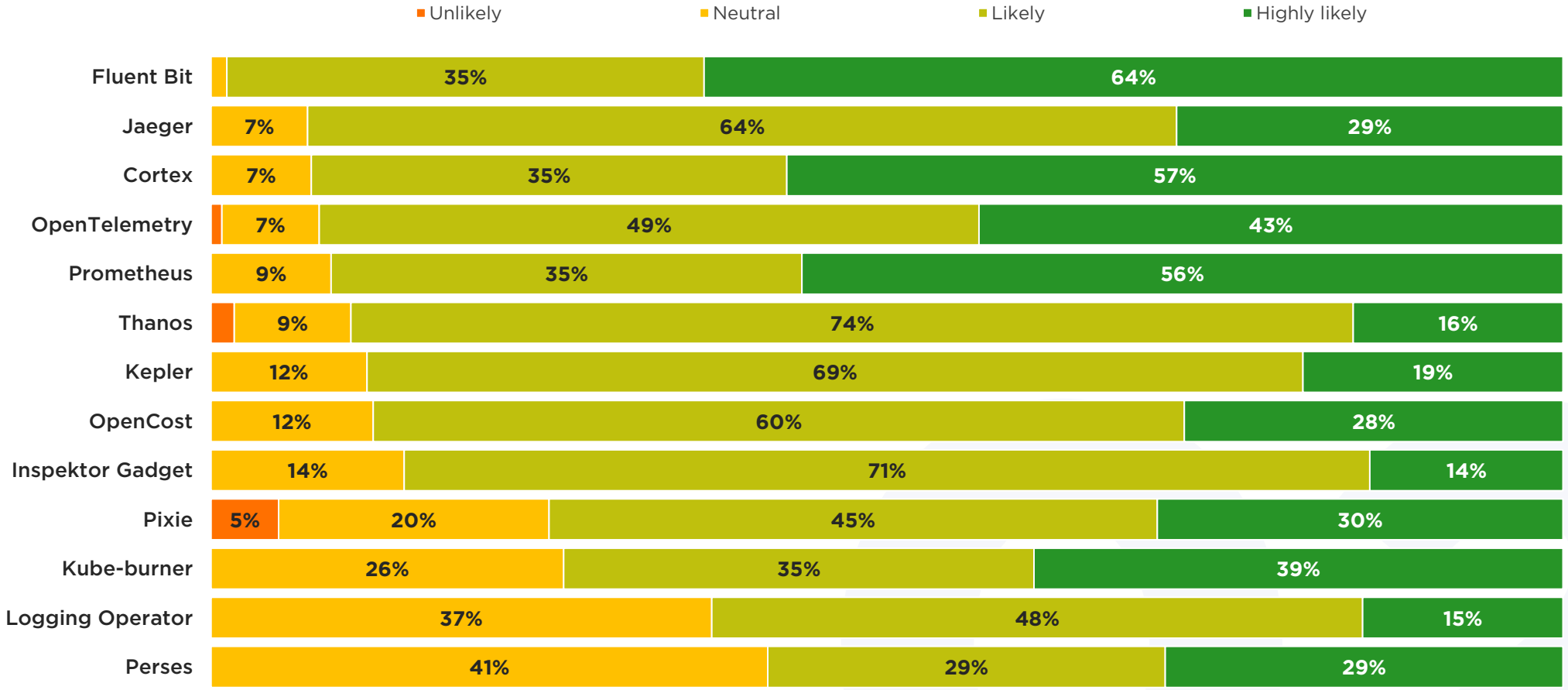
Fluent Bit received the highest recommendation ratings, with 99% of those familiar with it either being likely or highly likely to recommend it. Fluent Bit is not a small tool within the survey either, so this is reflective of a sizeable audience that universally recommends it. Most interestingly, Fluent Bit had maturity and usefulness ratings that rank in the middle of the pack. In particular, there are developers who only gave it a 3-star rating on either maturity or usefulness who then went on to be likely or highly likely to recommend it. A likely explainer for this recommendation lift comes from Fluent Bit being treated by many in the user base as a 'default choice,' even if logging is a less inspiring or interesting technology in their stack.

Behind Fluent Bit, the next highest recommendations were for Jaeger (93%), Cortex (92%), and OpenTelemetry (92%), though each reaches this position in a different way. Cortex is the most enthusiastically recommended of the three, with 57% highly likely to recommend it. Cortex has a specific purpose, providing scalable long-term storage for Prometheus metrics, and it received the highest maturity score of any observability tool. Its users, past and present, have likely sought it out for this purpose, and its mid-table usefulness score fits a tool that does one job well rather than many.

OpenTelemetry has by far the largest user base of the three, and its recommendations are broadly in line with its strong maturity and usefulness scores. As a competent tool with a broad range of uses across metrics, logs, and traces, it is recommended by a large audience without the gap between ratings and recommendation seen for Fluent Bit. Jaeger is the one that hides itself. While 93% would recommend it, this is dominated by those only likely to recommend it (64%), and ranked by highly likely alone, it sits joint eighth, in line with its eighth- and ninth-place rankings on maturity and usefulness. This suggests that while a large portion would recommend Jaeger, it is seen more as an additional utility than a tool developers are strongly advocating for.

Likelihood to Recommend Observability Tools

% of developers currently or previously using the technology (n=239)



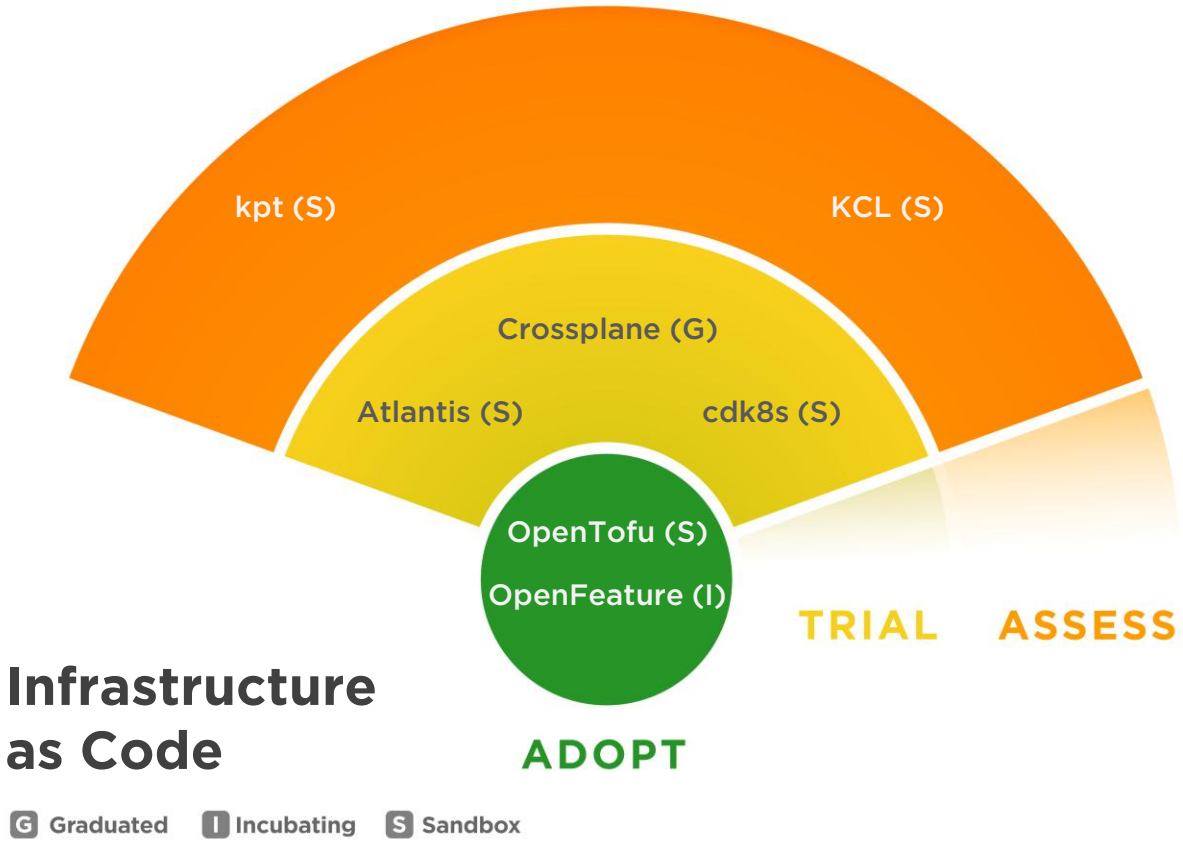
Question wording: How likely are you to recommend the following observability tools?

INFRASTRUCTURE AS CODE

03

3. Infrastructure as Code

OpenTofu and OpenFeature are the two technologies that are placed in the ‘adopt’ position of the application delivery technology radar. The ‘trial’ group for this technology category contains Crossplane, Atlantis, and cdk8s, with kpt and KCL in the ‘assess’ group.



Professional developers working with IaC technologies (n=195)

Based on developer perceptions: 'adopt' technologies are considered reliable choices for most use cases, 'trial' technologies are worth exploring to see if they meet your specific needs, and 'assess' technologies require careful evaluation before committing

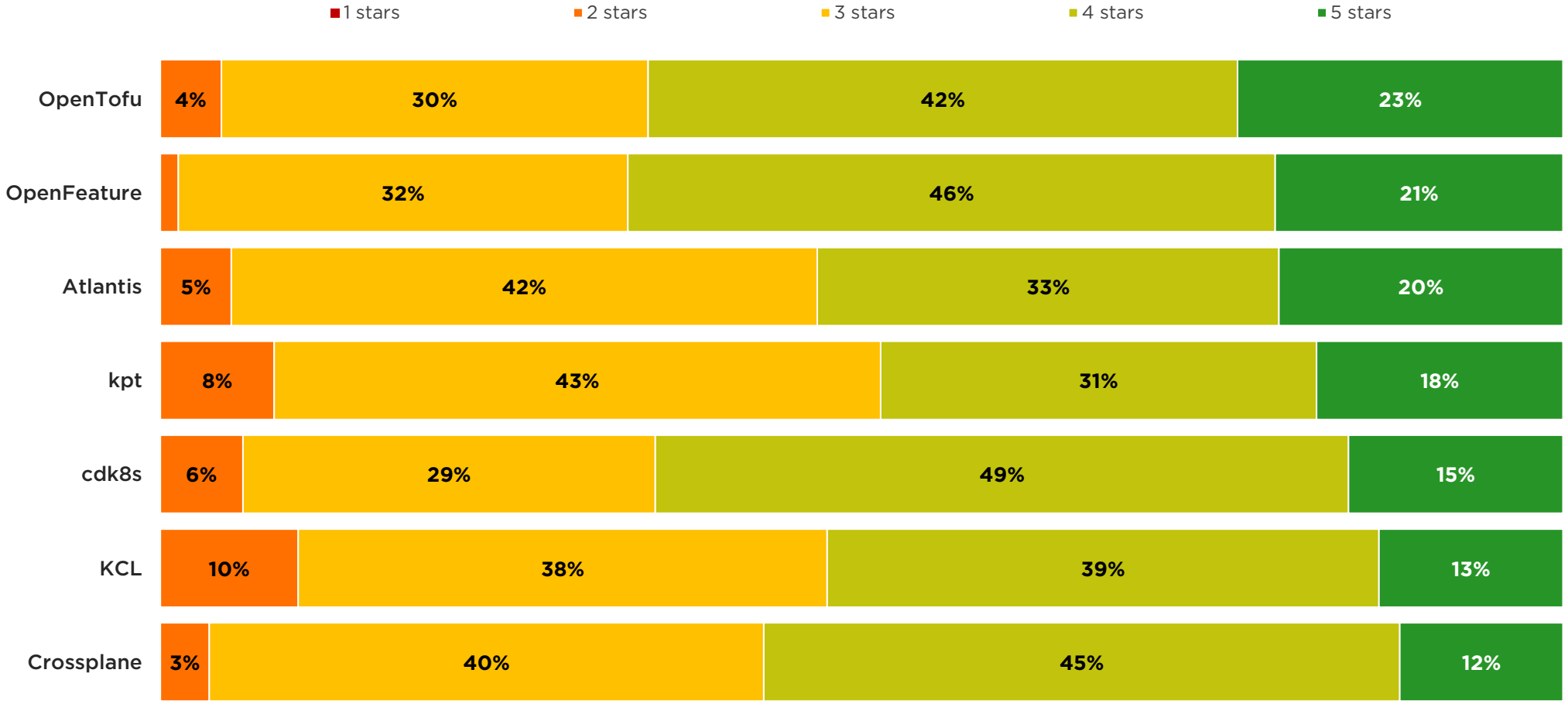
Maturity

OpenTofu and OpenFeature are closely grouped at the top for 5-star maturity ratings (23% and 21% respectively). OpenFeature has the largest proportion of positive ratings overall (67%), making it the most consistently well-rated tool for maturity. OpenTofu has a smaller overall positive rating proportion (65%), but these results are close.

The small differences likely come down to scope. OpenFeature is narrower, with a specific and well-defined scope as a standard for feature flagging, leaving less room for things to go wrong, and developers may respond to this as reliable and stable. OpenTofu, meanwhile, is broad in scope, but its inheritance of the Terraform codebase is likely a driver of developers resonating with it as a stable and reliable tool. This is notable given it is only a sandbox project within CNCF, suggesting developers are rating the maturity of the technology it carries forward rather than its standing as a project.

Maturity Ratings of IaC Technologies

% of developers currently or previously using the technology (n=195)



Question wording: How would you rate the following Infrastructure as Code technologies with respect to these aspects? (Maturity)

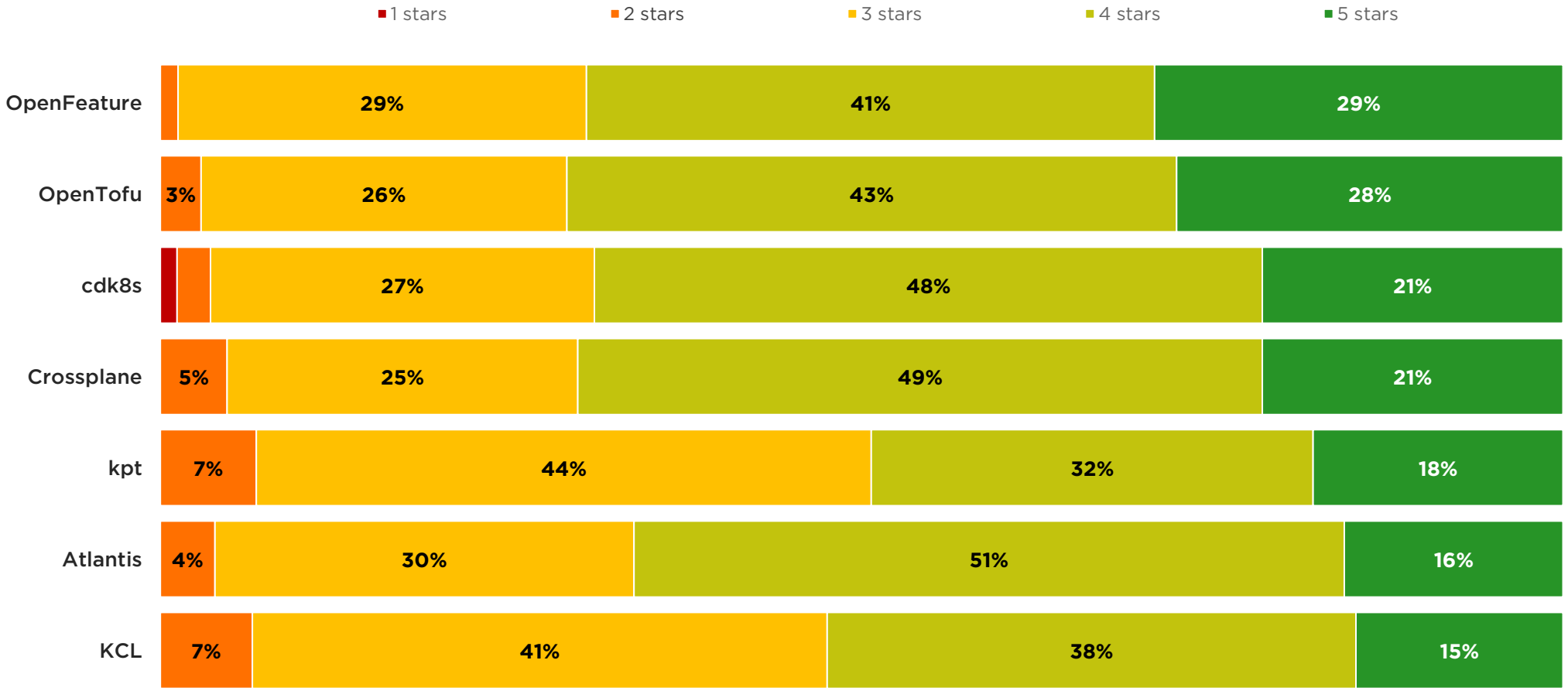
Usefulness

On measures of usefulness, OpenFeature and OpenTofu again lead, with 29% and 28% 5-star ratings respectively. For OpenFeature, its well-defined scope likely means it does one job well, while OpenTofu benefits from the ecosystem of providers and modules inherited from Terraform. cdk8s and Crossplane share the third spot with 21% 5-star ratings, despite ranking third from bottom and bottom, respectively, for maturity. Both tools offer developers a new approach: cdk8s' Kubernetes configurations written in general-purpose programming languages rather than YAML, and Crossplane using Kubernetes itself as a control plane for managing cloud infrastructure.

For cdk8s, the gap between usefulness and maturity is smaller than it may first appear, as its large share of 4-star maturity ratings would place it just behind the leaders on combined positive ratings. Crossplane is the clearer case of a tool whose approach developers value more than its current stability. The release of Crossplane v2.0 may be a contributor to this, with its utility assessed over the years of development, while maturity is based on the relatively recent major version release.

Usefulness Ratings of IaC Technologies

% of developers currently or previously using the technology (n=195)



Question wording: How would you rate the following Infrastructure as Code technologies with respect to these aspects? (Usefulness)

Recommendations

When considering whether respondents would recommend the tool, OpenTofu leads, with the highest proportions of both highly likely ratings (49% and 48%) and combined likely and highly likely ratings (96% and 93%). OpenTofu scored highly on both maturity and usefulness, and with the largest user base of any IaC project in this wave, its recommendation reflects a broad audience rather than a small enthusiastic group.

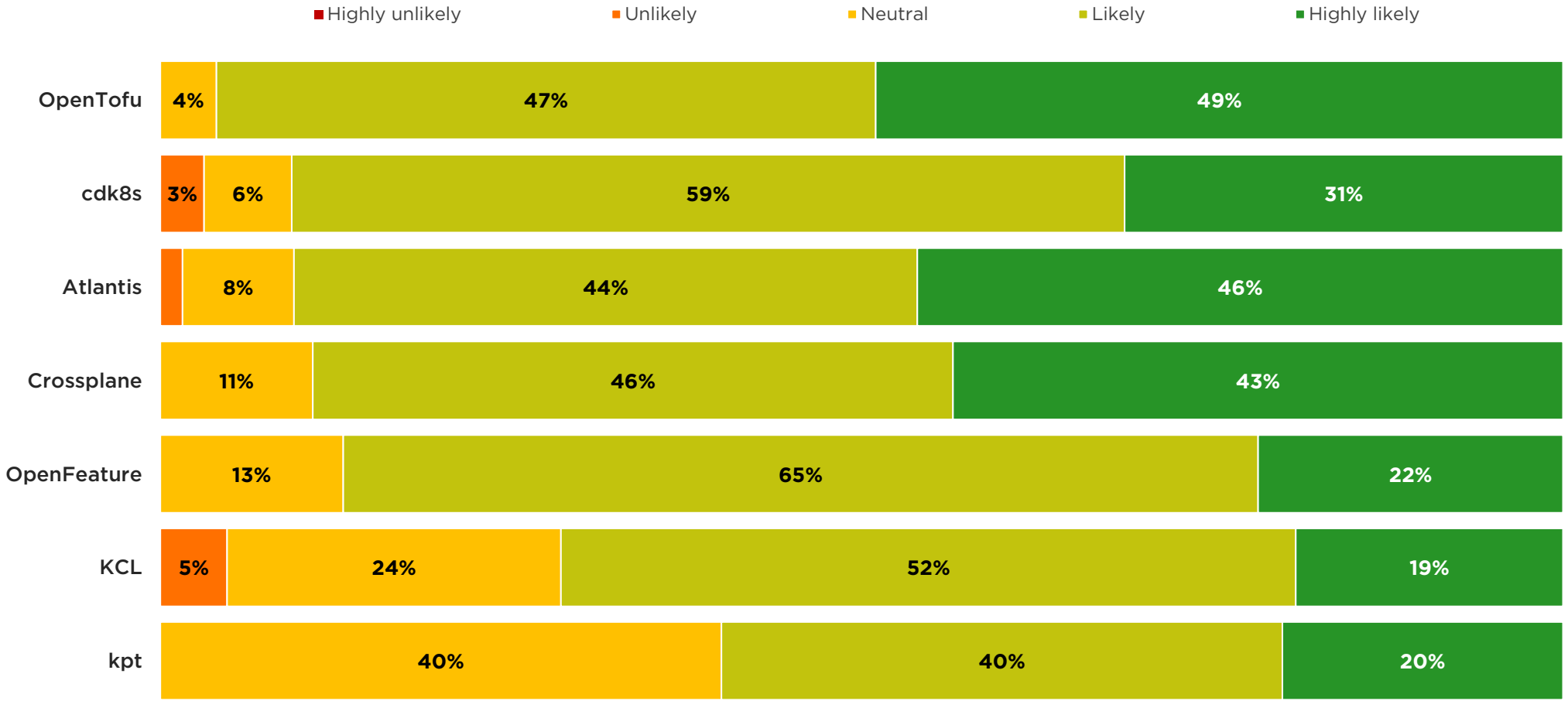
cdk8s is the second tool for recommendation when ranked on combined likely and highly likely ratings, but it falls to fourth on highly likely ratings alone. This matches what we see with maturity: consistently positive ratings but rarely top-shelf. This top-level recommendation may also be driven by respondents' perceptions of cdk8s's approach. Writing Kubernetes configuration in general-purpose programming languages rather than YAML suits some teams well, but not others, and developers may hedge their recommendation based on who they are recommending to.

OpenFeature follows a similar pattern to Jaeger in the observability section. Despite being a leader on maturity and usefulness, only 22% are highly likely to recommend it, though 87% would recommend it overall. Its narrow, well-defined scope as a standard for feature flagging makes it a dependable tool, but that same specificity makes it more of a useful utility than something developers enthusiastically advocate for.

KCL and kpt have the lowest recommendation on both measures. On highly likely ratings, they are trailing other tools, with 19% and 20%, respectively. Yet, the larger difference is on combined ratings, where KCL (71%) and kpt (60%) fall well below the next lowest, OpenFeature (87%). Rather than fewer developers being enthusiastic about them, a larger share withhold a recommendation entirely. Both tools ask developers to adopt an unfamiliar approach: KCL as a new configuration language and kpt through its configuration-as-data model. Unlike cdk8s, where the question is which teams the approach suits, here developers may be questioning whether the new approach is worth adopting at all.

Likelihood to Recommend IaC Tools

% of developers currently or previously using the technology (n=195)



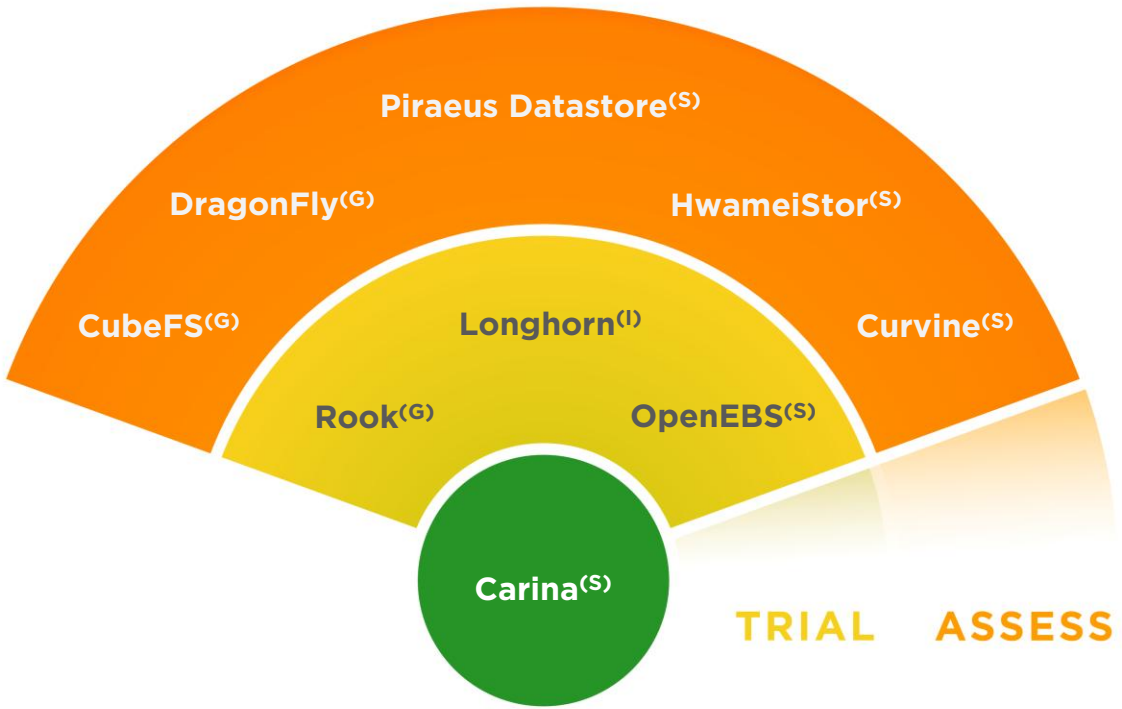
Question wording: How likely are you to recommend the following Infrastructure as Code technologies?

STORAGE TOOLS

04

4. Storage Tools

Carina is the only storage technology placed in the 'adopt' position, with Rook, Longhorn, and OpenEBS in the 'trial' position.



Storage ADOPT

G Graduated I Incubating S Sandbox

Professional developers working with storage technologies (n=279)

Based on developer perceptions: 'adopt' technologies are considered reliable choices for most use cases, 'trial' technologies are worth exploring to see if they meet your specific needs, and 'assess' technologies require careful evaluation before committing

Maturity

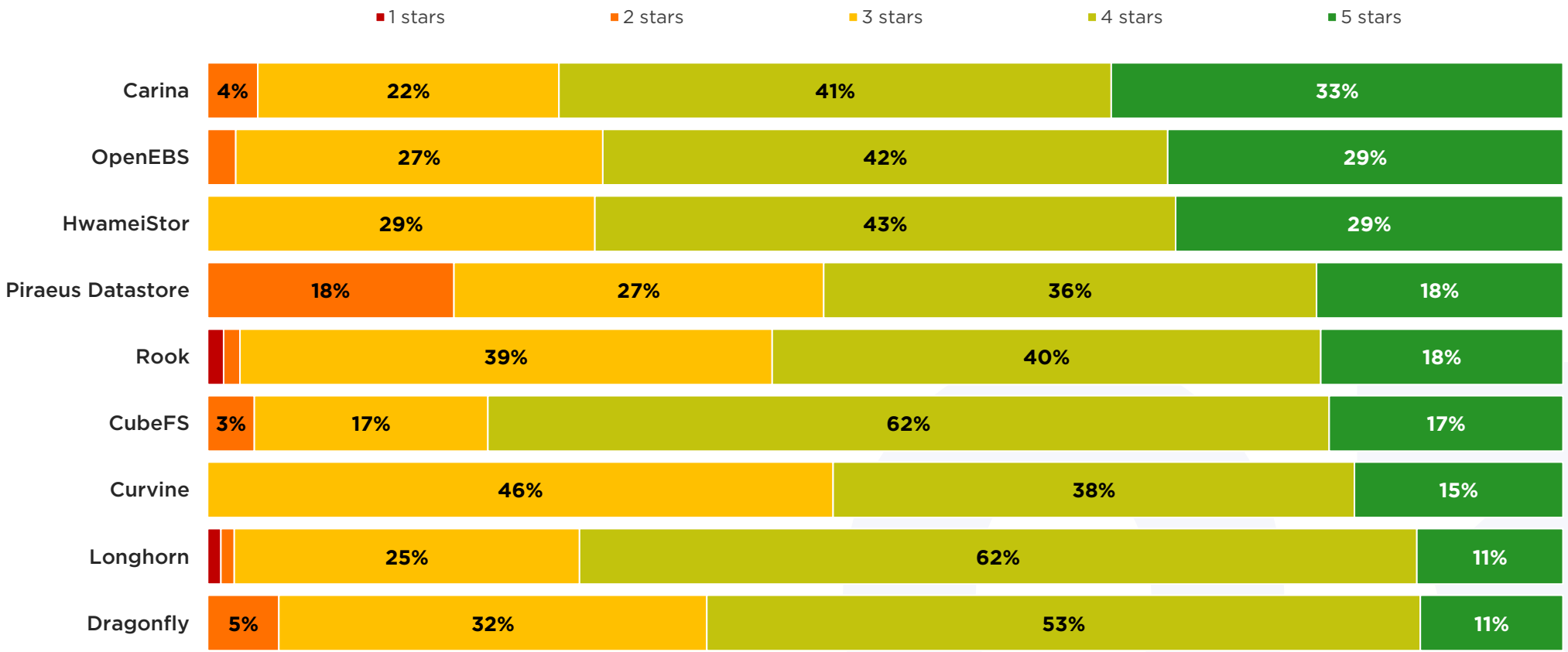
Carina leads on the proportion of 5-star ratings for maturity, at 33%. However, it is far from the leaders when combining 4- and 5-star ratings (77%), making the space somewhat divided on perceptions of maturity. Carina's division may be a reflection of what it does, as it being a local storage system for Kubernetes meaning it trades resilience for performance. Data held on a node's disk is only available as that node. Developers who choose it for performance and accept this trade-off may rate it highly, while those expecting the resilience of network-attached or replicated storage are likely to rate it lower.

At the other end, CubeFS[1] leads on combined 4- and 5-star ratings (79%), with Longhorn's combined positive ratings being comparable to the leaders (73%). There are tools with consistent positive ratings, but rarely reaching the peak. CubeFS and Longhorn have both been demonstrated with use in large-scale production, so steady positive ratings may fit their standing. As storage infrastructure, they are also likely judged by their absence of problems, which earns consistent positive approval more readily than enthusiasm.

[1] As seen in the methodology, the sample for this research had a low population of Chinese developers, a region that both CubeFS and Dragonfly have seen adoption and success

Maturity Ratings of Storage Technologies

% of developers currently or previously using the technology (n=279)



Question wording: How would you rate the following storage tools with respect to these aspects? (Maturity)

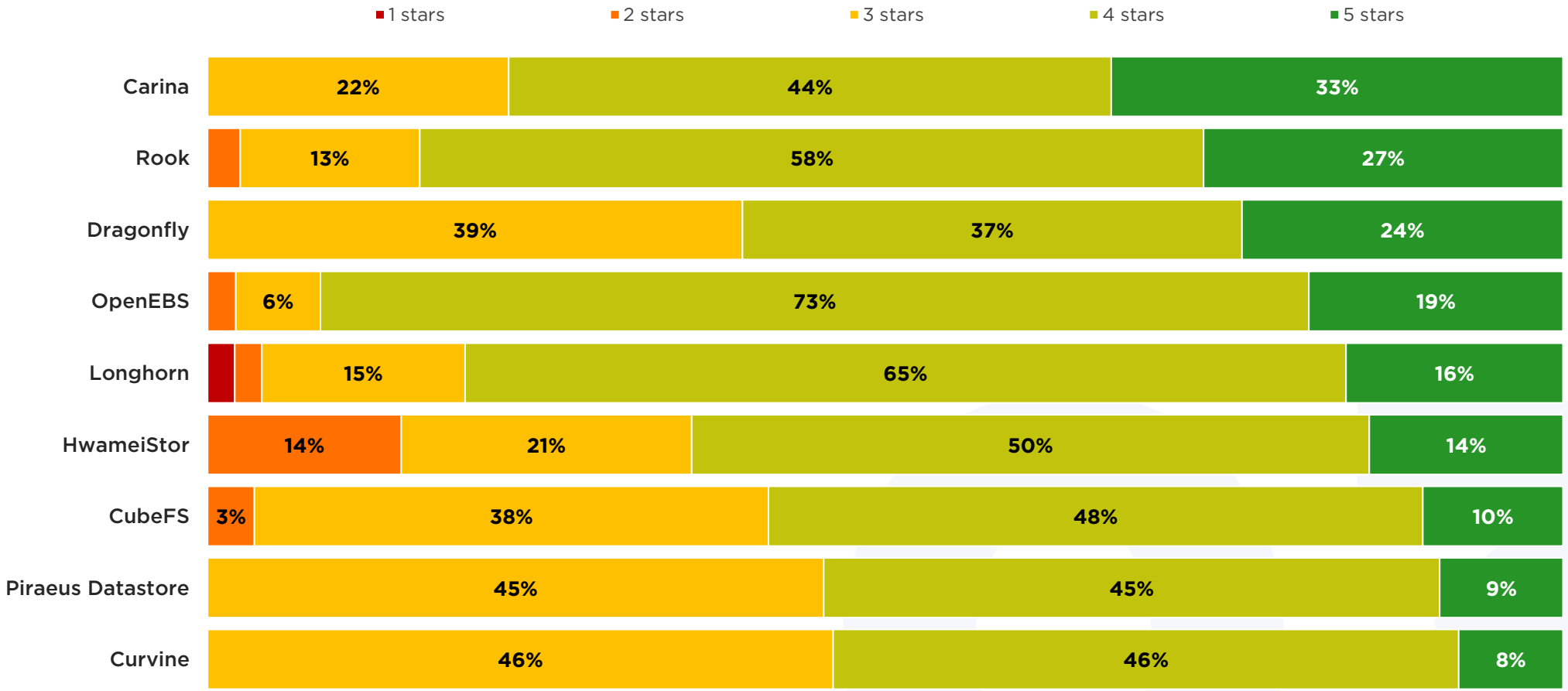
Usefulness

On usefulness, Carina is the highest rate tool, with 33% 5-star ratings (82%), with 18% giving it a 3-star rating. This mirrors its maturity ratings, suggesting a consistent split in how developers view the tool, while its strong top-end ratings across both measures support its placement in Adopt. The next highest rated tool is Carina, with 33% 5-star ratings, which is consistent with its strong maturity ratings. OpenEBS has the greatest proportion of combined 4- and 5-star ratings (92%), but with only 19% 5-star ratings, following the same pattern of consistent positive ratings but rarely the top shelf seen for maturity.

Rook and Longhorn are the technologies with the largest current and former user bases within our results, and both have much larger proportions of 4-star ratings (58% and 65%, respectively) than 5-star ratings (27% and 16%, respectively). As the most established storage options for Kubernetes, they draw broad approval rather than enthusiasm. Larger, more varied user bases also tend to produce more moderate ratings.

Usefulness Ratings of Storage Technologies

% of developers currently or previously using the technology (n=279)



Question wording: How would you rate the following storage tools with respect to these aspects? (Usefulness)

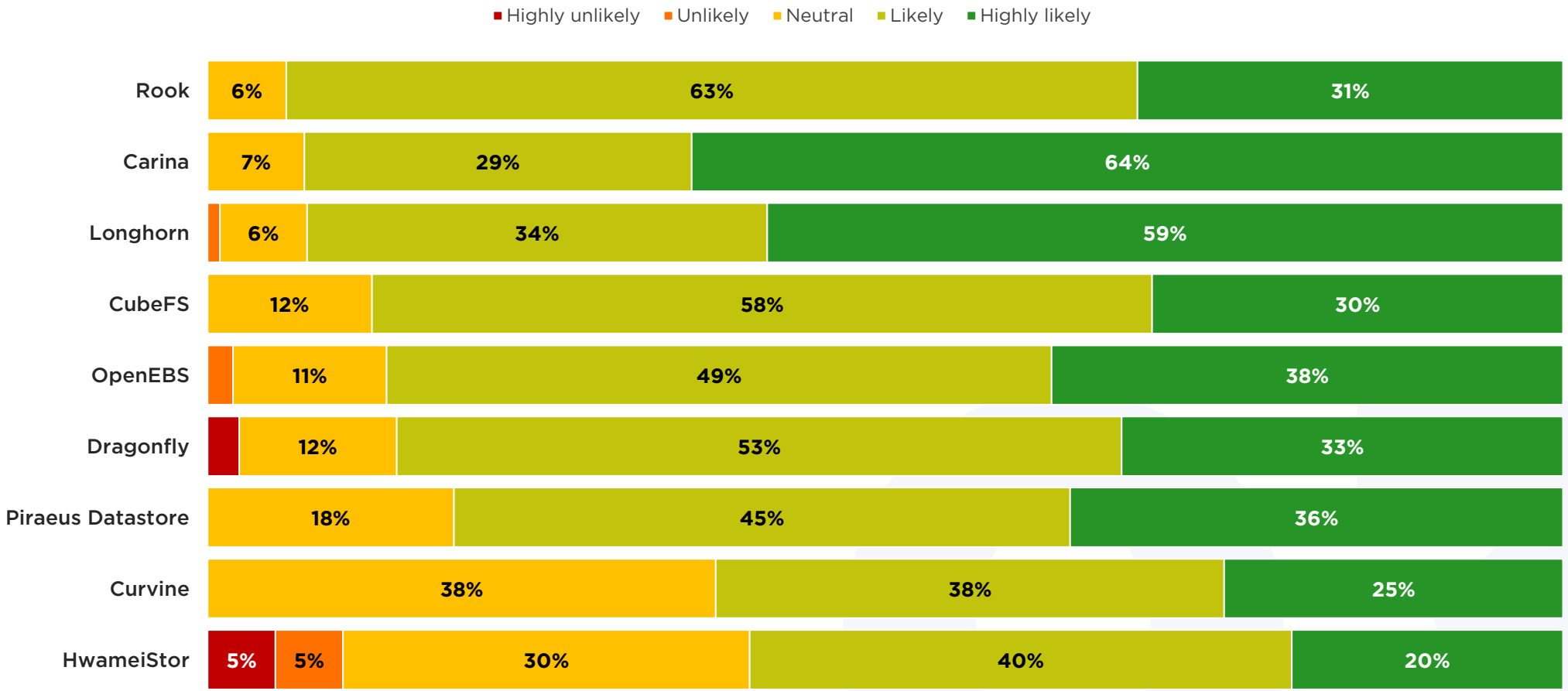
Recommendations

Rook receives the highest proportion of likely and highly likely to recommend ratings (94%), with Carina (93%), and Longhorn (92%) rounding out the leaders. Among the two most commonly used tools, however, there are two different responses. Longhorn may not have scored the highest on maturity or usefulness, but a large portion of its current and former users (59%) are highly likely to recommend it. Rook, on the other hand, is overwhelmingly recommended but with less enthusiasm, with its positive recommendation mostly driven by likely ratings (63%). A plausible explainer lies in what each tool asks of its users. Longhorn is generally considered simple to set up and run, making it an easy tool to recommend to almost any team running Kubernetes. Rook brings the power of Ceph to Kubernetes, but Ceph is notoriously demanding to operate. Developers may therefore recommend Rook with an implied caveat, hedging their recommendation based on whether the team they are recommending to has the expertise to run it.

Carina has a highly enthusiastic group, with 64% of current and former users stating they are highly likely to recommend it. This sits alongside its more divided maturity ratings, suggesting that even developers with reservations about its stability would still recommend it. As local storage for Kubernetes, Carina is well suited to performance-sensitive workloads such as databases, and for developers running those workloads, its trade-off of performance over resilience is likely the right one. Its recommendation may therefore reflect a clear sense of where it fits, rather than a view that it suits every team.

Likelihood to Recommend Storage Technologies

% of developers currently or previously using the technology (n=279)



Question wording: How likely are you to recommend the following storage tools?

PLATFORMS FOR HANDLING AI WORKFLOWS

05

5. Platforms for Handling AI Workflows

How an organization structures its developer platform is closely associated with how it approaches the integration of AI workflows. Two factors appear to matter: whether platform ownership is centralized in a single team and whether that team builds its own platform or assembles third-party tools. Organizations with centralized platform teams tend to commit to a single approach, while those with more distributed ownership tend toward hybrid approaches, such as a separate experimentation environment but shared production deployment. To explore this, we examined how the prevalence of different platform ownership models among surveyed developers interacts with the approaches organizations are taking to handle AI workflows within those platforms.

The proportion of developers who report their organization has a platform team that builds a custom internal developer platform (IDP) is small, just 9%, but these developers are predominantly either building a separate dedicated platform for AI (38%) or extending their existing platform to support AI (33%). Only 5% take a hybrid approach. As full owners of their IDP, this likely represents a deliberate decision to commit to one direction. This is most evident when compared with those who have multiple teams collaboratively building platform capabilities or individual teams choosing and managing their own tooling, where 43% and 54% respectively have a hybrid approach.

5. Platforms for Handling AI Workflows

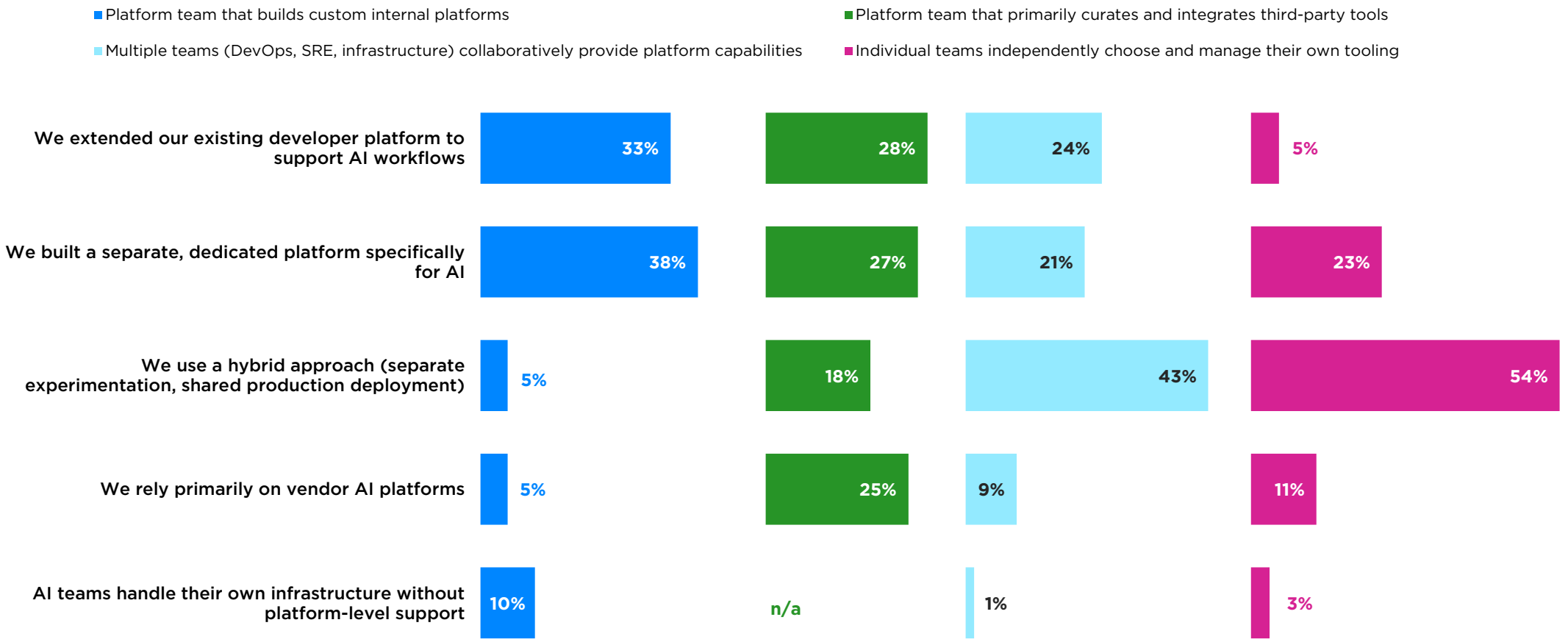
Extending the existing platform to AI workflows is also much less common in these groups, with 24% of those with collaborative platforms doing so. For individual teams handling their own tooling, the figure is just 5%, though this partly reflects that these organizations often have no shared platform to extend. This suggests that extending platform capabilities to manage AI workflows is more challenging without a single organization-wide platform and when responsibility is not concentrated within one team.

Building a dedicated separate platform is a popular approach for all groups but is much more common among those with a single team building custom platforms (38%) than among platform teams curating third-party tools (27%), multiple teams working together (21%), and individual teams managing it themselves (23%). This likely reflects capability: building and running a separate platform requires platform-engineering capacity, which teams building their own IDP already have.

Organizations whose platform team primarily curates third-party tools see their workflows spread across extended (28%), separate (27%), hybrid (18%), and relying on AI vendor platforms (25%). Their reliance on AI vendors is the highest of any group, which is logical given these teams already rely on third-party tools. However, like teams building custom platforms, they have a much lower propensity for a hybrid approach than organizations with distributed ownership. A hybrid approach creates a seam between environments that someone must maintain. A central platform team owns both sides of that seam and has a reason to avoid it, whereas in organizations where ownership is spread across teams, hybrid approaches may emerge from teams' separate choices rather than a single decision.

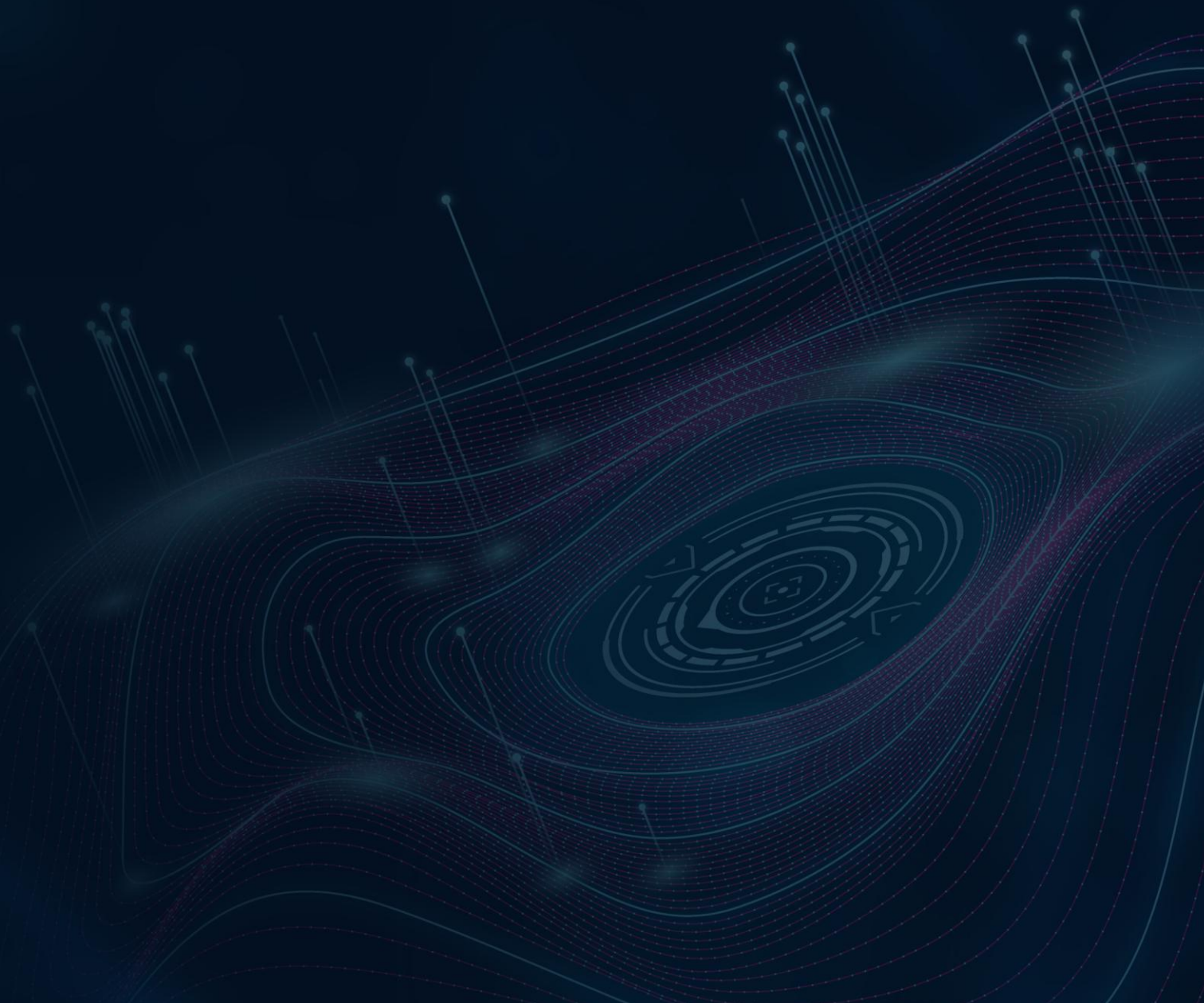
Approaches to AI Workflows by Internal Platform Approaches

% of developers integrating AI workflows with each organizational approach to developer platforms (n=364)



Question wording: Which of the following best describes how your organization approaches internal development platforms and tooling? | Which of the following best describes your organization's approach to AI workflows?

METHODOLOGY



Subjective nature of Likert scales

In our research, we employed Likert scales to capture developers' opinions on the maturity and usefulness, from 1 to 5 stars, of the various observability, IaC, and storage technologies surveyed. While these ratings are inherently subjective, reflecting individual perceptions and experiences, they provide valuable insights into the developer community's views. The nature of our research is centered on investigating developer **perceptions** of these aspects, making the subjective nature of the ratings not only acceptable but also valuable for our analysis. Although the subjective nature of Likert scales may influence the interpretation of results, as different respondents may have varying standards for rating, this variability enriches our understanding of the developer experience.

Despite these nuances, analyzing the distribution of ratings — such as the difference between the number of 5-star ratings and those of 1 and 2 stars — serves as a practical measure for understanding developer sentiments. This approach allows us to identify trends and patterns that can inform decision-making, highlighting areas of strength and opportunities for improvement within the surveyed technologies. Thus, we assert that Likert scales are an effective tool for gauging developer perceptions and experiences.

Respondent demographics

Respondents were initially asked about where their projects ran or were deployed, to identify their position as a 'cloud developer.' Following this, they were asked which technologies they were currently using that we associate with cloud native development approaches, including technologies such as Infrastructure as Code, service meshes, and serverless computing.

Respondents were recruited from SlashData's proprietary Developer Nations panel as well as select third-party panels. For privacy and data minimization purposes, exclusion is based on internal consistency and survey-taking behavior metrics. As such, information on the organization the respondent works for is not carried through to any analysis. This privacy also helps encourage greater honesty from respondents, who do not have concerns that their expressed opinion will be associated with them.

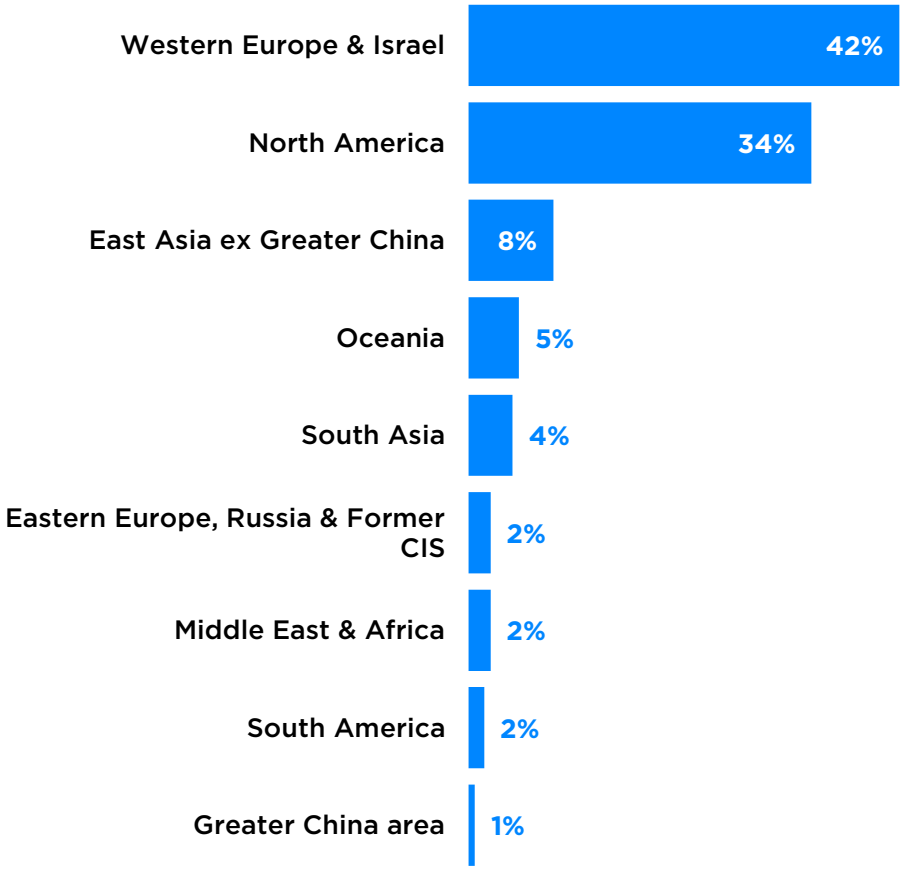
Due to the nature of third-party panels making up the significant majority of respondents, we consider the risk of multiple respondents from the same organization responding to be low and, as such, do not engage in deduping cleanses. However, should more than one individual from the same organization respond to the survey, we do not consider it to impact the validity of the results.

Within the same organization, developers may be using different technologies. Further, while usage was used in the determination of each technology's position on the technology landscape radar, the developer's personal perceptions corresponded to 75% of the score the technology received.

Methodology

Region

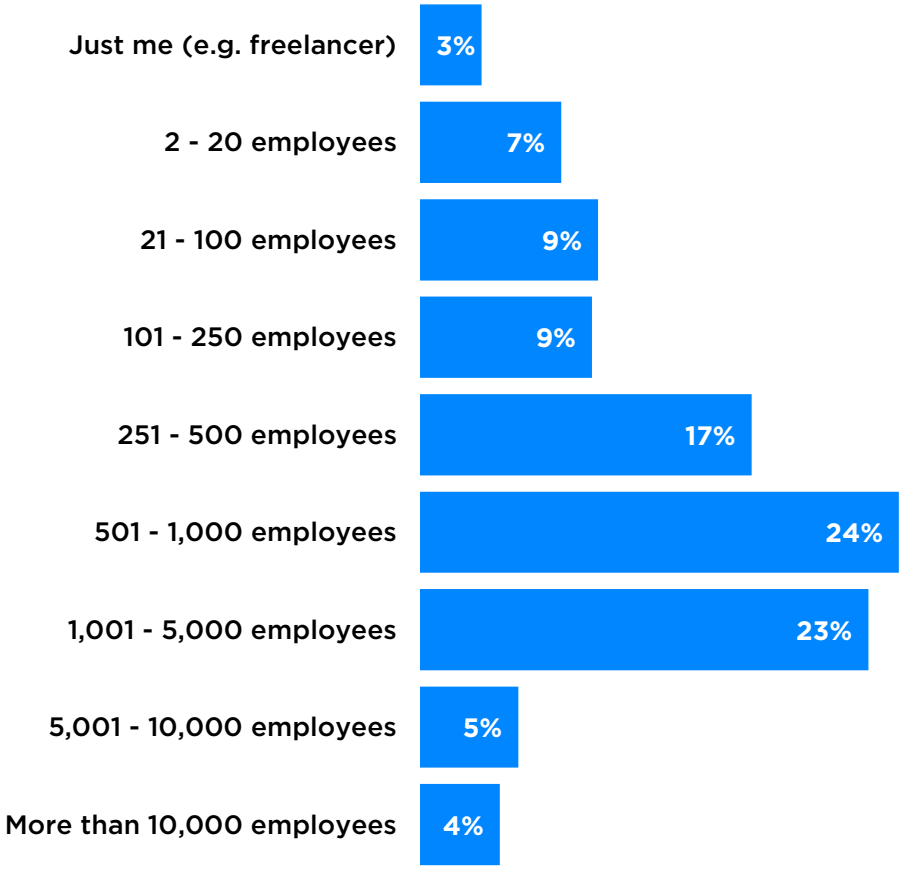
% of developers (n=324)



Question wording: Which country are you based in?

Organization Size

% of developers (n=324)

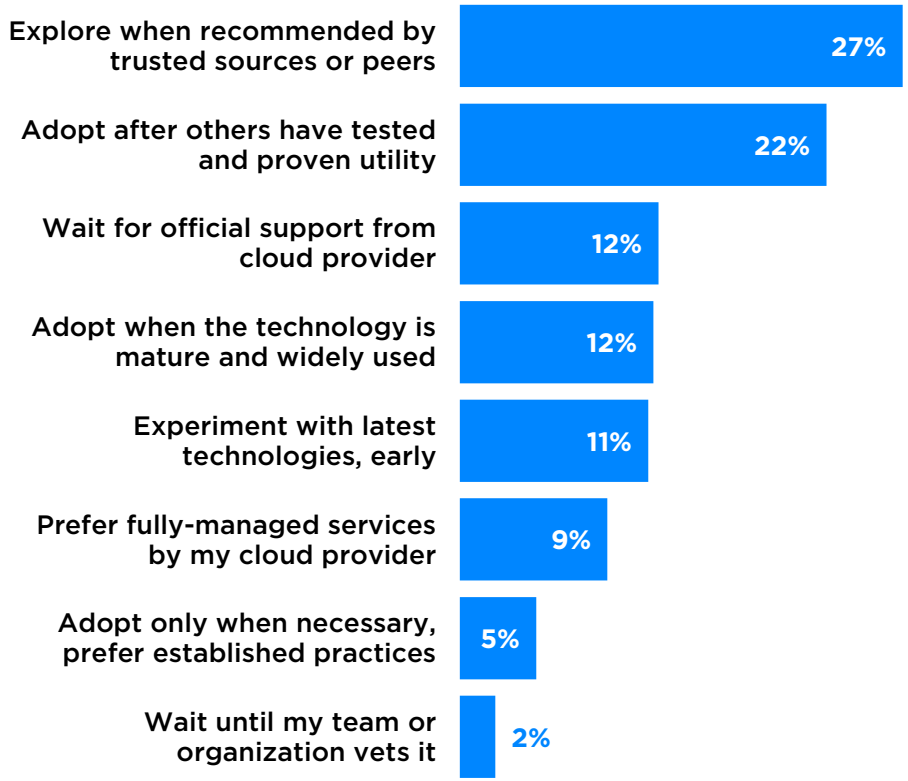


Question wording: What is the size of the organization that you work for?

Methodology

Approach to cloud native technology experimentation and adoption

% of developers (n=420)



Question wording: Which of the following best describes how your organization approaches internal development platforms and tooling? | Which of the following best describes your organization's approach to AI workflows?

Industries and sector involvement

% of developers (n=324)



Question wording: Which industry best describes the organization that you work for?

Navigate AI technology decisions with confidence & clarity

SlashData is an AI analyst firm which has been working with the top Tech brands to provide clarity and confidence in their decision-making.

For 20 years, we have been tracking software technology trends and helping technology brands make product and marketing investment decisions, challenging assumptions and reframing market trends to empower industry leaders to drive the world towards the future.

Find us at slashdata.co




/DATA
CLOUD NATIVE
COMPUTING FOUNDATION