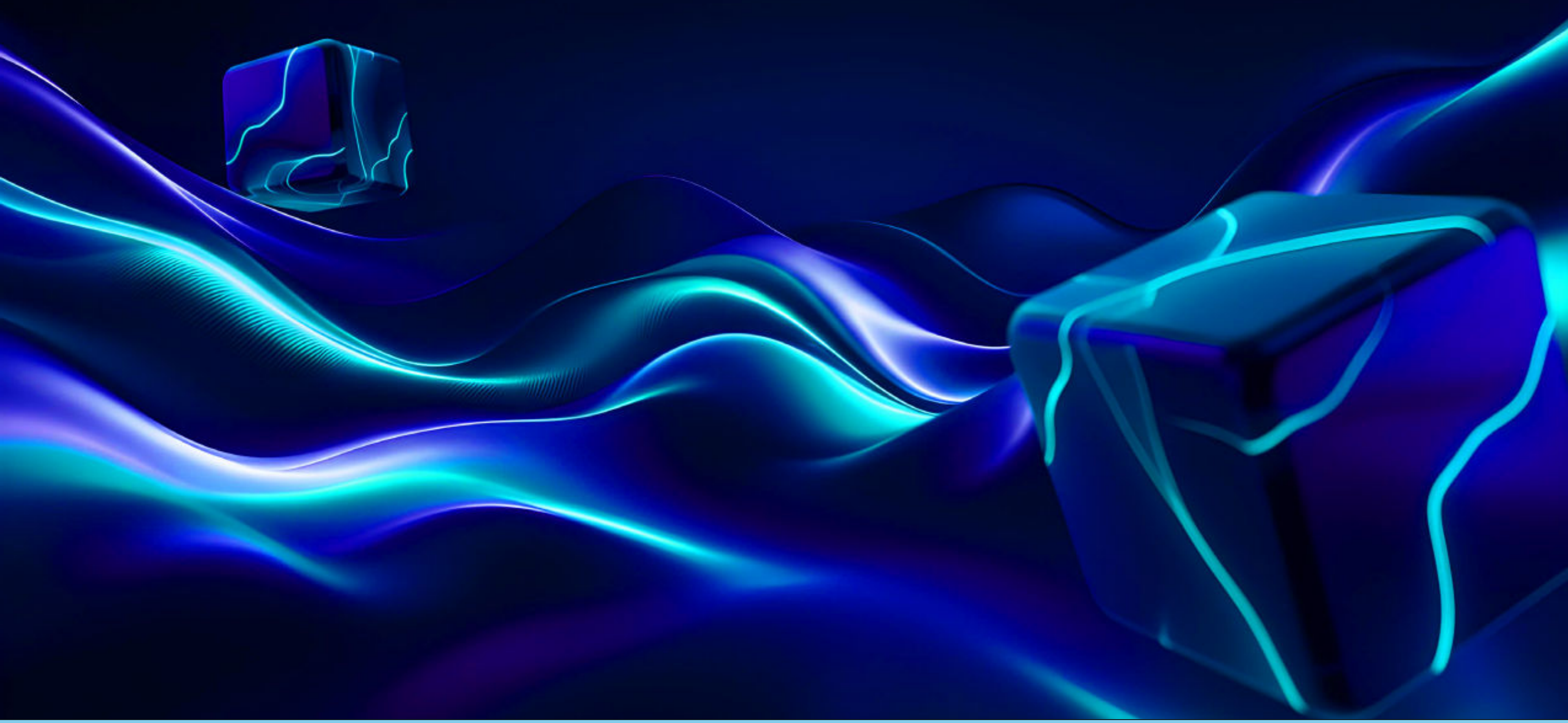
Achieving Reliable Releases in the Era of Cloud-Native Delivery

Day 1 Release Survey Research Report



Paul Nashawaty
Principal Analyst | theCUBE Research

September 2025

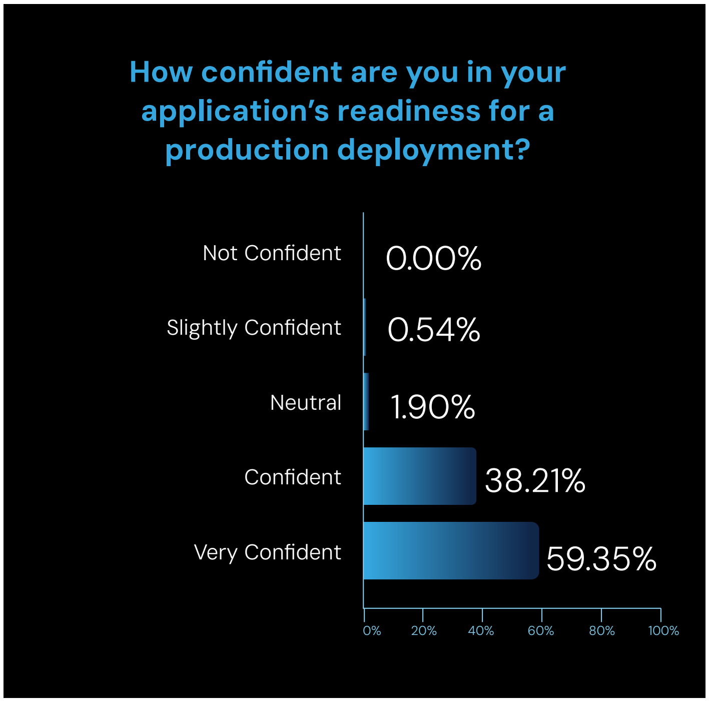
Executive Overview

In theCUBE Research's Application Development and Modernization Day 1 Release survey, we explore how organizations plan, execute, and monitor production deployments. The results show a wide range of maturity, with some teams running highly automated, low-disruption pipelines, and others still struggling with readiness validation, environment provisioning, and post-release visibility.

A clear theme is present: automation drives confidence. Teams that have integrated observability, infrastructure-as-code, and mature CI/CD are more resilient, release, and recover more quickly. This report analyzes the key areas of Application Readiness, Infrastructure and Environment Readiness, Deployment Process, and Monitoring & Observability, highlighting both the challenges and the practices that set high-performing organizations apart.

Application Readiness

Application readiness is the foundation of a successful release. Early and frequent code and performance validation reduces post-release problems and boosts stakeholder confidence. 59% of respondents to our 2025 survey expressed they were "very confident" in the production readiness of their application. The remaining individuals fell into the "neutral" or "confident" categories. These confidence gaps are often caused by inadequate end-to-end validation, insufficient integration testing, or a lack of coordination between QA and development teams.

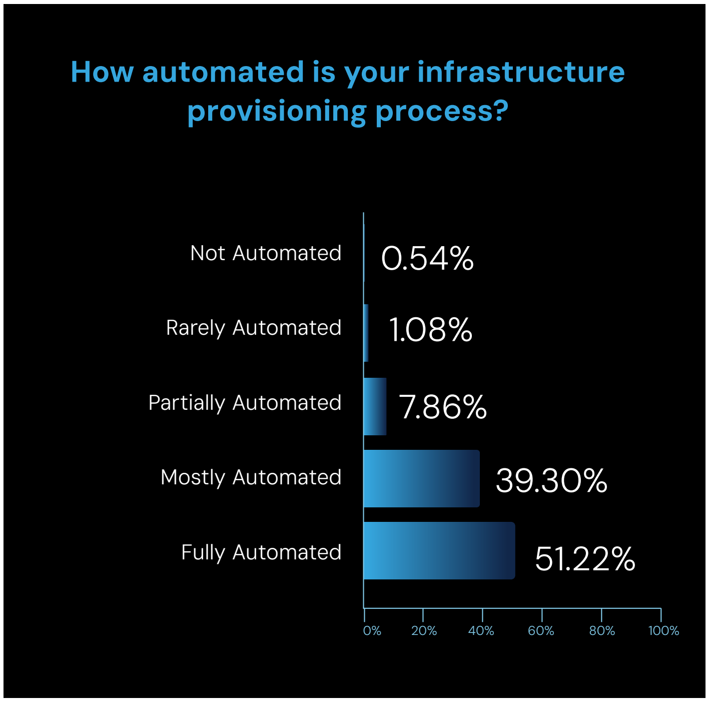


How confident are you in your application's readiness for a production deployment?

Embedding automated regression tests, performance benchmarks, and security scans into release pipelines helps catch issues earlier. This supports a "shift-left" approach, where testing is pulled closer to development, shortening feedback loops and reducing fire drills right before release.

Infrastructure & Environment Readiness

Even the most polished application can fail if the environment isn't ready. Here, automation plays a crucial role. In our 2025 survey, 51% of participants reported having fully automated infrastructure provisioning, while another 39% indicated that their process was mostly automated. As a result, a sizable minority still relies on ad hoc templates or manual setup, which increases the likelihood of configuration drift, staging/production mismatches, and release failures.

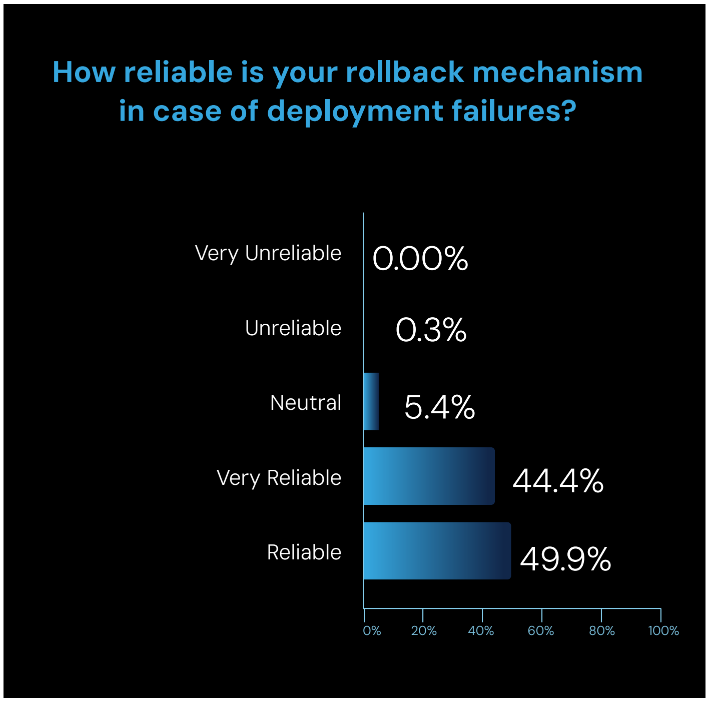


How automated is your infrastructure provisioning process?

Teams that use standardized baselines and Infrastructure as Code (IaC) report fewer delays caused by the environment and faster incident recovery. In faster-moving organizations, self-service provisioning portals further reduce bottlenecks, giving developers and testers direct access to the resources they need without waiting on centralized teams.

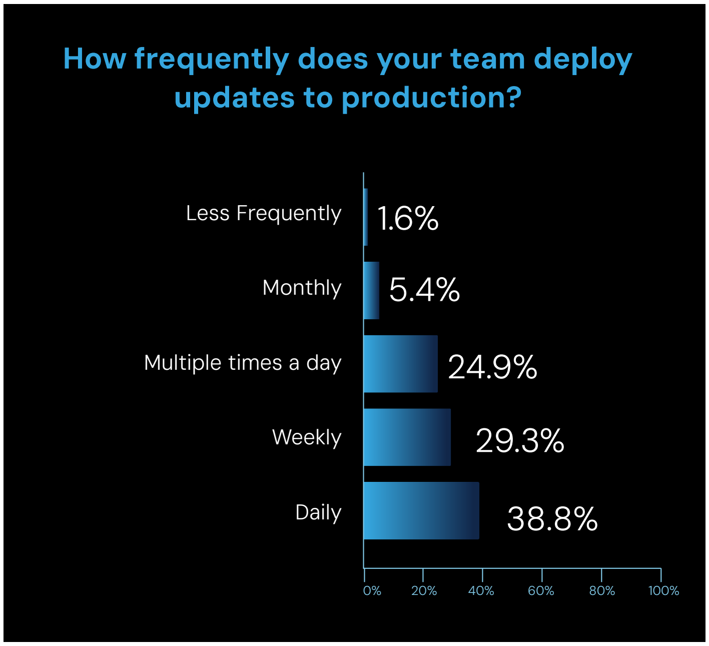
Deployment Process

The data shows growing adoption of release strategies designed to minimize disruption. Blue/green and canary deployments are increasingly common, which enable incremental rollouts and rapid rollbacks when problems arise.



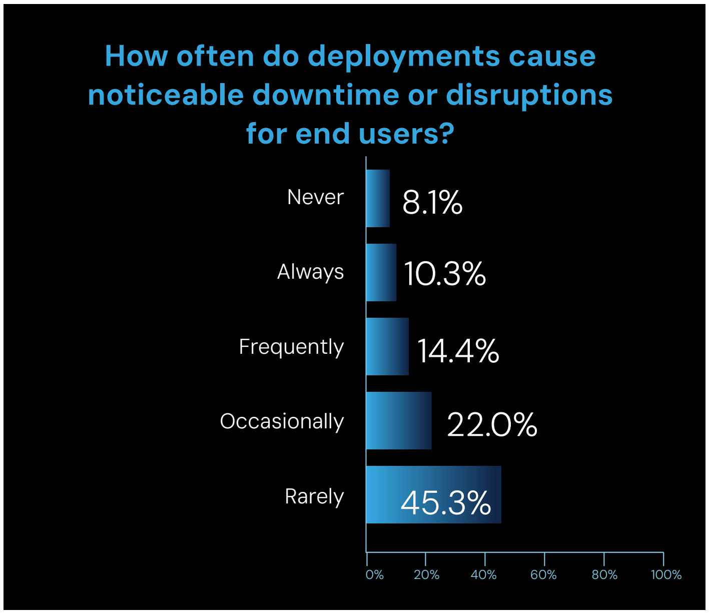
How reliable is your rollback mechanism in case of deployment failures?

With almost 50% of respondents rating their mechanisms as "reliable" and another 45% as "very reliable," rollback reliability is comparatively high. However, dependability is just one component. The frequency of deployments still varies greatly, ranging from monthly or less frequent updates in more conventional settings to multiple daily releases in mature DevOps shops. Lower frequencies tend to raise risk, as larger, bundled changes carry a greater potential impact.

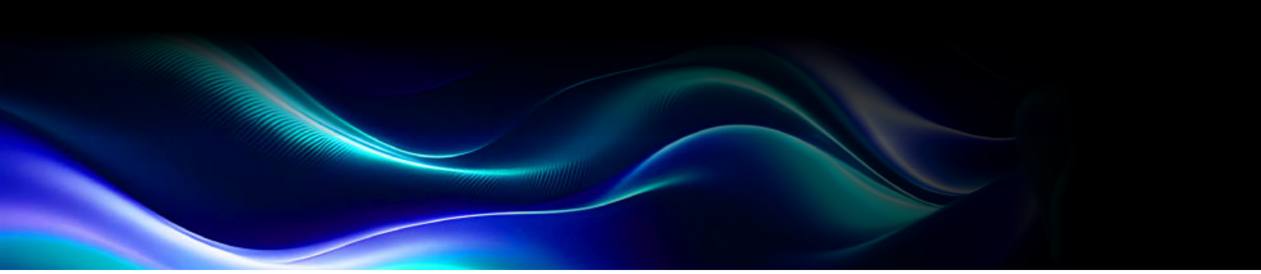


How frequently does your team deploy updates to production?

Downtime remains a challenge. 10% encounter downtime nearly every release, 14% encounter it regularly, and 45% say it happens infrequently. Stronger tooling and a shift in culture (from automated rollbacks and pre-deployment checks to a blameless postmortem culture that encourages continuous improvement) are needed to reduce incidents.



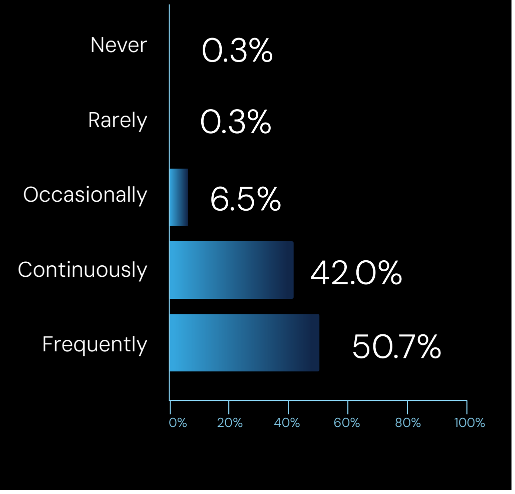
How often do deployments cause noticeable downtime or disruptions for end users?



Monitoring & Observability

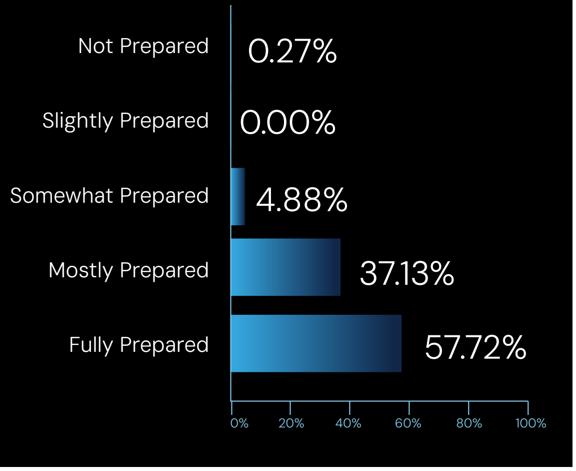
What happens after a release is just as important as the release itself. Teams can identify and address problems early with the help of efficient monitoring and observability, safeguarding user experience and uptime. Progress has been made, according to 2025 survey results, with 50% of organizations regularly and 42% of organizations continuously monitoring the health of their applications. However, a sizable portion still operates with limited visibility after deployment.

How frequently are monitoring dashboards used to assess the health of the application post-release?



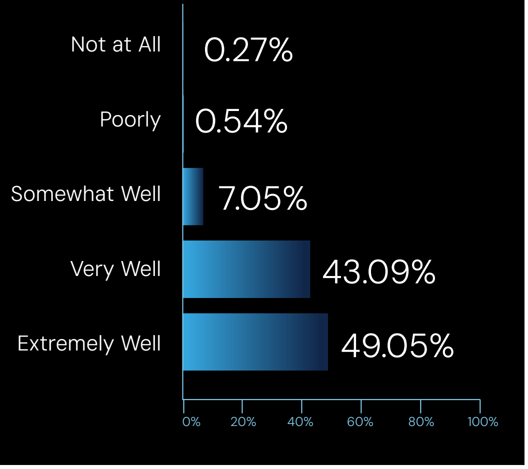
The efficacy of real-time monitoring tools is not entirely clear. Proactive alerts, integrated dashboards, and a strong relationship between releases and performance are advantageous to specific teams. Disjointed tools, data silos, or shaky links between deployments and results still constrain others.

How prepared is your application for monitoring and observability upon release?



Effective teams automatically connect deployment events with baselines and incident reports, integrating observability into the release process. This reduces mean time to recovery (MTTR) and expedites root cause analysis. With predictive alerts and anomaly detection assisting teams in resolving issues before they impact end users, AI-assisted monitoring holds even more promise for the future.

How well does your log management system support troubleshooting and root cause analysis?



Closing the Day-1 Release Gap: Building Confidence in Cloud-Native Delivery

For years, teams have chased the same goal: ship faster without breaking things. But according to the Day 1 Release Survey by theCUBE Research, many organizations still find that speed and confidence rarely travel together. Fifty-nine percent of respondents said they feel "very confident" in production readiness—yet nearly a quarter still encounter downtime during releases. Reliability, it seems, is less about how often teams deploy and more about what they can see when they do.

Automation Everywhere—But Visibility Lags Behind

Cloud-native teams have made enormous progress in automation. Over half of survey participants report fully automated infrastructure provisioning, and another 39 percent are close behind. Kubernetes clusters spin up and down by the minute; infrastructure-as-code is the default, not the exception.

Yet automation alone doesn't guarantee stability. When ephemeral services change faster than humans can track, the question isn't whether pipelines run—it's whether teams can trust what's running. Without a clear link between change events and system health, every release becomes a small leap of faith.

From Rollback to Insight

The research highlights a paradox. Rollbacks are nearly universal and highly reliable—about 94 percent of teams rate their rollback mechanisms as "reliable" or "very reliable." But downtime still occurs for roughly 25 percent of organizations on a frequent basis. The ability to undo change is not the same as the ability to understand it.

True resilience depends on shortening the time between "something changed" and "we know why." That means integrating release context—version tags, feature flags, configuration changes—directly into observability data. When metrics, traces, and logs are correlated with each deployment, teams can detect deviations in golden signals within minutes, not hours, and recover before customers notice.

The Role of Observability in the Cloud-Native Feedback Loop

What happens after a release is just as critical as the release itself. The survey shows that most organizations now monitor their applications either frequently (50.7 percent) or continuously (42 percent). However, many still operate with limited visibility across silos.

Modern observability practices close this loop. They connect telemetry with the continuous delivery pipeline so every rollout becomes a data point for learning. Instead of asking "did it deploy?", teams ask, "did it improve the user experience?" The most mature organizations treat observability as part of their production strategy, not a bolt-on after thought.

Confidence Through Context

High-performing teams in the study share a common thread: they embed observability directly into their automation workflows. Automated validation, self-service provisioning, and contextual monitoring reduce mean-time-to-recovery and improve stakeholder trust. These practices transform Day 1 operations from reactive firefighting into proactive assurance.

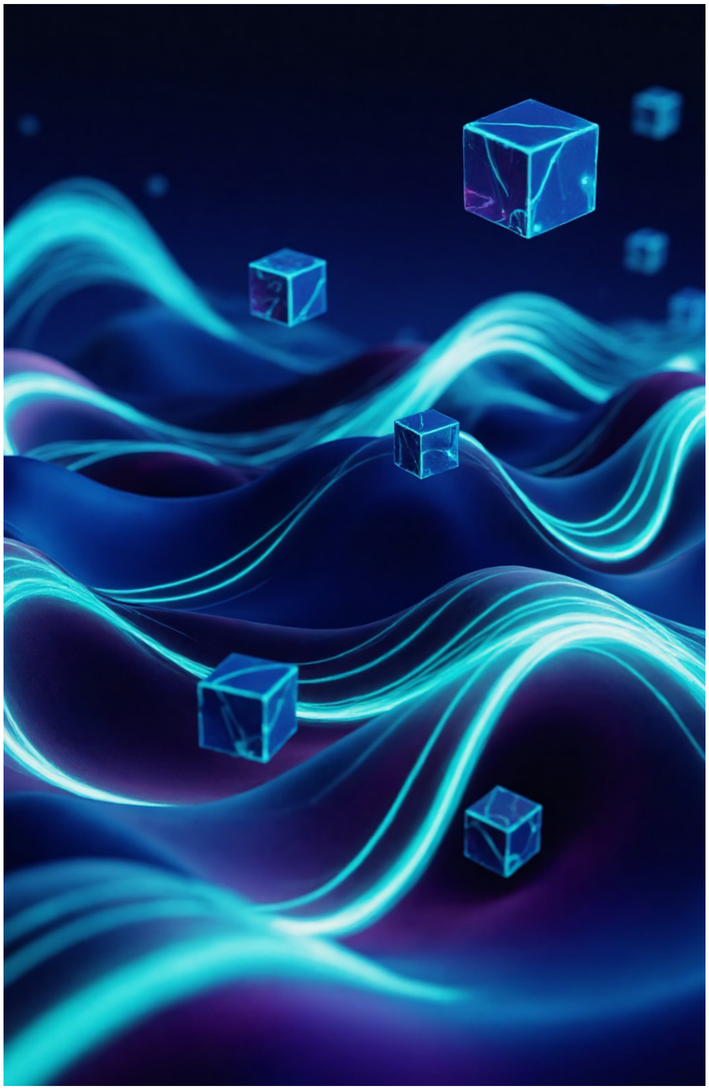
This isn't about more dashboards or alerts—it's about building a living understanding of how systems behave over time. When observability keeps pace with change, confidence stops being a feeling and becomes a measurable outcome.

Looking Ahead

AI-assisted monitoring is on the horizon, promising predictive alerts and faster anomaly detection. But the report makes clear that no algorithm can substitute for sound observability foundations: clean telemetry, deployment awareness, and cultural maturity around continuous improvement.

The future of cloud-native delivery will belong to organizations that connect automation and insight. As the survey concludes, success in Day 1 operations will hinge on one thing above all—strengthening the connection between releases, monitoring, and continuous improvement.

In other words, the next era of reliability isn't just about deploying faster. It's about seeing faster, learning faster, and turning every release into a moment of clarity.



Conclusion

Our 2025 survey has reinforced the critical role of automation, observability, and cultural maturity in achieving reliable, low-disruption deployments. While many organizations now embrace strategies such as Infrastructure as Code, blue/green rollouts, and real-time monitoring, uneven adoption continues to create risk through downtime, configuration drift, and post-release blind spots. High-performing teams stand out by embedding automated validation, enabling self-service provisioning, and linking observability directly to deployment pipelines; all practices that reduce MTTR, accelerate recovery, and build stakeholder confidence. Moving forward, success in Day 1 operations will hinge on scaling automation, normalizing modern deployment strategies, and strengthening the connection between releases, monitoring, and continuous improvement.
To dive deeper into the full dataset, benchmarks, and practitioner insights behind this research, please reach out to theCUBE Research for direct access and tailored advisory services.

Disclaimer

All trademark names are the property of their respective companies. Information contained in this publication has been obtained by sources theCUBE Research, a SiliconANGLE Media company, considers to be reliable but is not warranted by theCUBE Research. The publication may contain opinions of theCUBE Research, which are subject to change. This publication is copyrighted by theCUBE Research, a SiliconANGLE Media company.

Contact

Silicon Valley
989 Commercial Street
Palo Alto, CA 94303

Boston Metro
95 Mount Royal Avenue
Marlborough, MA 01752

David Butler
david.butler@siliconangle.com
774-463-3400