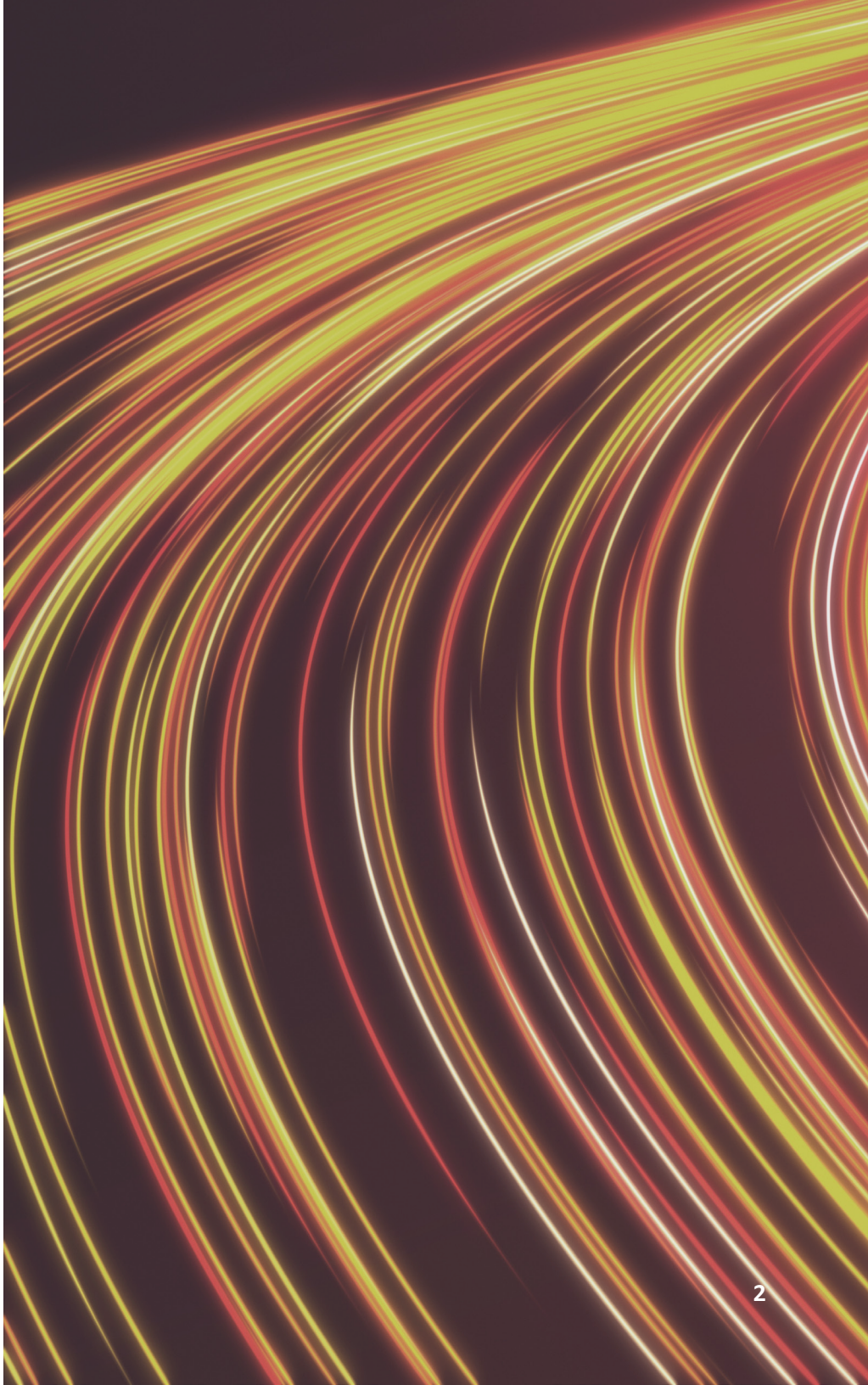

aws modern apps
Adopting a Modern Dev+Ops model

Modern Dev+Ops is allowing organizations to meet the evolving needs of their customers faster and more consistently by closing the gap between dev and ops functions.

DevOps has changed the world. Amazon and the internet changed how software was delivered, from a disk in a box bought in a store to available digitally as a service built on AWS. But it was DevOps that changed the update cycles of that software from months or years to days and brought Dev+Ops closer together.

Companies of all shapes and sizes have tossed aside waterfall product management cycles and embraced agile and DevOps for the promise of faster innovation, improved security, performance, and resilience, as well as happier developers and customers. But even with all of the benefits of DevOps, it isn't perfect. DevOps came into existence before developing and hosting applications in the cloud was a standard practice, and though it has been around for a long time, it is still largely open to interpretation from one organization to the other. For example, many organizations still consider DevOps to be a dedicated team, and in other cases, it can mean developers doing all of the operations work. As technology has evolved, so has our definition of DevOps. We call this approach Modern Dev+Ops, and it is centered around bringing developers and operations closer by sharing



operational tasks like compliance, observability, resilience, and infrastructure earlier into the development process and enhancing it with AI/ML.

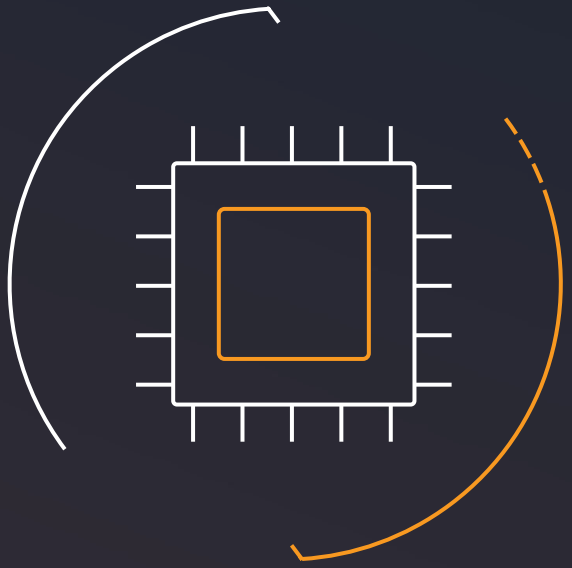
The *State of DevOps Report* states that companies who adopt DevOps and move their deployment frequency from weekly/monthly to hourly/daily improve their lead time from months to days and their change failure rate from 46–60 percent to 0–15 percent. For many customers, success in the cloud is not just whether they have adopted on-demand, scalable infrastructure but also at least partly a function of how they transform their development and operations practices. Now, customers such as Coca-Cola Argentina, 3M, Lululemon Athletica, and The Washington Post have transitioned to a Modern Dev+Ops approach, enabled by AWS Dev+Ops services to bring their development teams even closer to their operations teams to further expand on the benefits of DevOps.

"Using AWS, we've gone from deployments taking six weeks to one per week, and very soon we expect that to be multiple deployments per day."[1]

—Rick Austin, Manager of Advanced Technology, 3M Health Information Systems

"Instead of taking two days to build a new production account, we can do it in minutes using AWS CloudFormation templates and AWS CodePipeline. That means we can launch small-scale projects that cost very little and take very little time to set up. With that agility, we can experiment and get to the best solutions rather than having to settle for what we have resources for. Using AWS, we can push out new features and applications much faster than we could previously."[2]

—Sam Keen, Director of Product Architecture, Lululemon Athletica



[1] *https://aws.amazon.com/solutions/case-studies/3M-health-information-systems/*

[2] *https://aws.amazon.com/solutions/case-studies/lululemon-athletica/*

Teams with higher evolved DevOps practices

77%

restore services after an incident in less than a day

60%

fully remediate security vulnerabilities in less than a day

3x

more effective at change management

Many customers have not yet adopted Dev+Ops but are turning to it in search of a solution that can help them deliver faster and with fewer errors. Other customers have started their Dev+Ops journey but have struggled to achieve the desired level of speed and success they envisioned. In many cases, the problem customers face is that they find it too hard.

1. It's too hard to get started. It can seem daunting. There's a lot to learn and a lot to do to implement change.
2. The industry solutions that currently exist do not sufficiently consider enterprise needs: security, compliance, high availability, access control, etc. As a result, DevOps teams spend a lot of time building a solution only to need to replace it down the line.
3. Many industry solutions do not do enough to make Dev+Ops easy for DevOps teams. For example, many solutions are reactive and not proactive. They tell a customer when something is already going wrong rather than help prevent the issue in the first place with advanced warnings and suggestions for how to best proceed.

Changing culture is the hardest part of Modern Dev+Ops. Getting everyone on the same page takes leadership, time, and commitment. As developers begin to participate in more of the responsibilities that were managed by operations teams in the past, there needs to be a sense of shared responsibility. The following is a set of ways of working, both cultural and behavioral, that, when adopted, provides a mechanism for building a high-performing Dev+Ops organization.

This eBook covers the guiding principles AWS has identified in high-performing Dev+Ops organizations:

- Practice accountability
- Measure yourself
- Make incremental improvements
- Automate everything
- Codify wherever possible
- Implement belts and suspenders
- Standardize tools

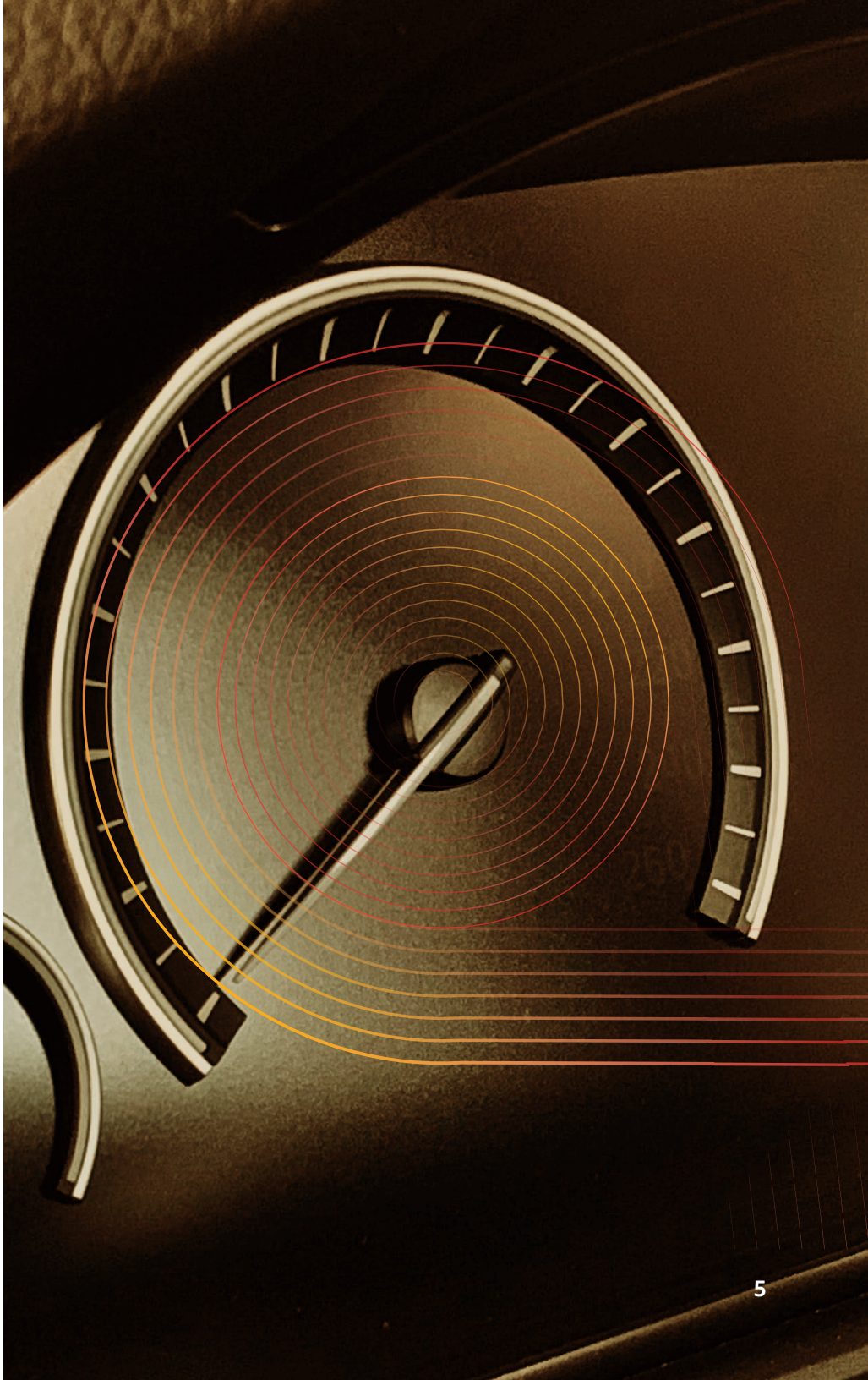
Practice accountability

The goal of Dev+Ops may be speed in software delivery, but the initial intent was to bring development and operations closer together so they can collaborate on solving problems. While organizational structures have varied over time, the common solution has been to focus on a culture of accountability. Teams and individuals can have specializations and areas of expertise, but they must share accountability in tackling and solving problems.

Measure yourself

The ultimate goal of Dev+Ops is to go faster. But teams frequently don't know where to start. You can start by measuring yourself. This means measuring your current time to release. Which parts of your software delivery process are taking the longest and why? Can we improve? What other key performance metrics are important to your success? Are there areas of your process that are siloed between dev and ops? Start by measuring the inputs and results of those metrics, and you'll find places where you can focus and demonstrate results that build confidence within your team.

- Time to release
- Bottlenecks or blockers in the pipeline
- Release velocity



Make incremental improvements

Nobody can go from zero to a high-performing Dev+Ops organization in a matter of days or weeks. It's important to identify some guiding principles and then start small, making steady but incremental improvements. Similarly, you won't be able to speed up your software process if you keep yourself tied to big changes and big releases. It's much easier for humans to process small, incremental changes with a limited number of new things to understand than it is big, large, complex changes.

Automate everything

The goal of Dev+Ops is to keep everything in continuous motion. Automation reduces the time and errors associated with human intervention. There are many ways to add automation to your Dev+Ops workflow, but the most basic one is by adding in a Dev+Ops CI/CD pipeline with automated tests, deployments, and rollbacks.

Codify wherever possible

Another part of the trick of automation and Dev+Ops is to codify things such as your infrastructure and policies so that both humans and machines can understand them. The most basic use case is infrastructure as code (IaC), which automates the provisioning of your cloud infrastructure as defined by you, giving you a blueprint that helps you keep track of changes and share best practices across an organization. Another example is policy as code, which enables central teams to write rules that are enforced by automation.

Infrastructure-as-code files include:

- **Compute**
- **Storage**
- **Identity, Access, Security**
- **Application resources**

Implement belts and suspenders

Ensuring best practices are followed is about both encouraging and enforcing them. Encouraging best practices can be done by the use of best practices templates that are shared around an organization, typically with infrastructure as code. Enforcing best practices can be done with processes such as policy as code. But the next frontier for belts and suspenders is using rules and artificial intelligence to find potential errors before they affect users and giving Dev+Ops teams guidance on how to implement the best practice.

Standardize tools

Having standardized tools enables organizations to scale. Writing policies and ensuring best practices is much harder, if not impossible to enforce across a heterogeneous set of tools. Using tools from fewer vendors that have been designed to work together can help, but ultimately more standardization saves time and improves errors.

Why AWS?

AWS is in a unique position to help customers move faster with their cloud transformation by enabling them with a core set of Modern Dev+Ops practices, tools, and training. With AWS, customers can realize the benefits of the cloud faster and accelerate their innovation.

1. **Amazon is a pioneer in Dev+Ops and innovation**

   Amazon has pioneered many of the best practices that are common in Dev+Ops and has over a decade of experience working with high-performing customers to provide solutions that fit their needs. AWS services incorporate Amazon best practices by default.

2. **AWS has the broadest and deepest portfolio of integrated Modern Dev+Ops services**

   From the core monitoring and CI/CD services to infrastructure as code to the latest services that use big data and AI to provide proactive advice and recommendations, AWS has all the services you need to build a high-performing Dev+Ops team or organization.

3. **Security is job zero; availability is job one**

   AWS Dev+Ops services are built with security best practices by default. For example, your CI/CD pipelines are isolated per project by default so it's easy to control access permissions for only those who need it. Additionally, AWS services are all highly available so teams can be productive.

4. **Built for both the developer and organization**

   Many developer tools neglect to think about the needs of the organization. AWS focuses on the needs of both the developer and the organization and makes it easy to implement governance and compliance controls that enable development teams to go fast, safely.

Action plan

So how do you bring Modern Dev+Ops to life on AWS? The following are recommended best practices that organizations of all shapes and sizes have used to enact the guiding principles from above:

GitOps

GitOps is an approach where infrastructure as code is hosted in a git repository and follows the same merge request process as application software code. GitOps is designed to eliminate any out-of-band changes to an application's infrastructure. Infrastructure as code is the practice of defining your application's infrastructure in a code format, often with a template or file, and automating the provisioning of your infrastructure to match that file definition. With GitOps, the infrastructure definition file is hosted in a git repository and each commit follows a merge request process like application software code and flows through continuous integration and continuous delivery (CI/CD) automation. In the case of GitOps, infrastructure files are tested and infrastructure provisioning is automated with rollbacks to the previous state in the case of errors.

AWS provides a complete GitOps solution built around AWS CloudFormation. AWS CloudFormation lets developers and operations teams easily define, provision, and manage their infrastructure-as-code files in console with common infrastructure languages like YAML or JSON or with AWS Cloud Development Kit (AWS CDK). AWS CDK lets developers define CloudFormation templates with common programming languages like Python, TypeScript, C#, Java, and JavaScript. CloudFormation has a first-class integration with AWS CodePipeline, which allows for automated CI/CD workflows and integrates with popular version control services like AWS CodeCommit, GitLab, Bitbucket, and GitHub. When a new infrastructure definition file is created, it is hosted in a version-controlled environment. As changes are made to the CloudFormation template, CI/CD automation is triggered, and developers can easily deploy or roll back/forward their changes.

"As Rivian grows at a rapid pace, we need a highly scalable system. Changes that took five days now occur within minutes."[3]

—Surendra Balu, 3DExperience Technical Lead, Rivian

[3] *https://aws.amazon.com/solutions/case-studies/rivian-case-study/*

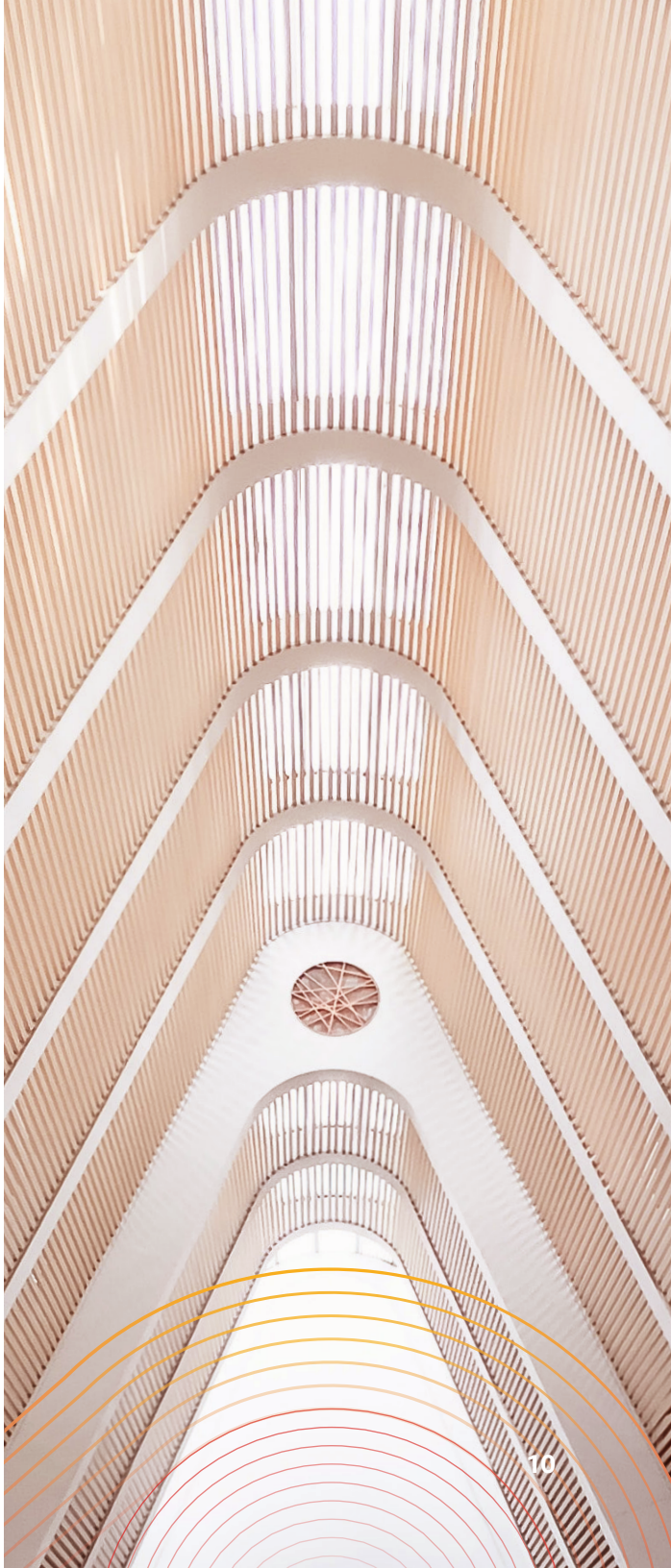
Continuous integration and continuous delivery

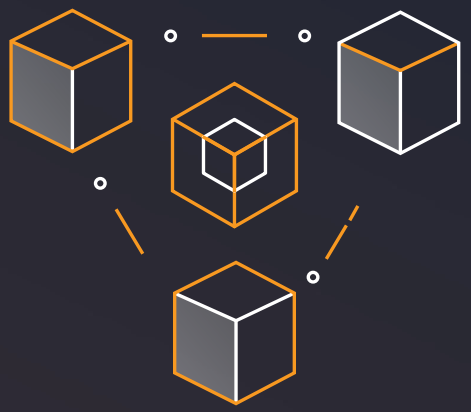
CI/CD is the foundational best practice of automating the preparation of software for release using pipelines that cover build, test, and deployment. CI/CD helps teams go faster and reduces errors by taking out error-prone manual processes and removing the need to babysit software releases. For customers with longer release times, CI/CD solves this problem by embracing continuous change to a system. Engineers write tests that detect potential errors or violations in code and reject the update back to the teams. In the event a bug is shipped to production, teams can automate rollbacks to a stable version and maintain uptime.

AWS provides a fully managed solution for building CI/CD in the enterprise or startup. AWS CodeBuild, AWS CodeArtifact, AWS CodePipeline, and AWS CodeDeploy are a set of connected Dev+Ops services purpose-built to help customers implement CI/CD. The services take lessons from Amazon's internal CI/CD tools and encourage best practices by default. For example, by isolating each project within a single CodePipeline pipeline, security-conscious customers can easily lock down permissions to only those who are supposed to have access to the project, removing the need to worry about vulnerabilities in a shared build server. Additionally, AWS makes CI/CD easier and requires less expertise. Services such as Amazon CodeGuru Reviewer can help in the pull-request phase of the software release cycle by augmenting code reviews with machine learning that helps identify critical issues, security vulnerabilities, and hard-to-find bugs during development. Developers receive guidance and can improve their code by resolving issues before new features get deployed into production. Additionally, AWS Proton is a service platform engineering teams can use to pre-configure all the different tools needed for infrastructure provisioning, code deployments, monitoring, and updates for their development teams.

"With AWS CodeBuild, our application build now takes about 10 minutes; using Jenkins it used to take up to an hour. To get the same performance on Jenkins would cost four times as much because we would need to spin up 50 Jenkins instances to have the builds finish that quickly."

—Bryan Kane, Senior Engineer, Coursera





AIOps

AIOps is the shift toward more automation and more proactive mechanisms that enable teams to innovate faster with confidence. Designed to reduce the amount of knowledge needed by the developer and augment their experience by leveraging AI in the Dev+Ops workflow, AIOps can provide useful insights before problems arise, help teams be proactive, enforce best practices by default, and ultimately innovate faster. For example, developers can detect deviation from best practices, concurrency issues, and other common coding bugs before they've had a chance to affect a system and get actionable recommendations that will improve system performance.

With AWS machine learning models informed by two decades of Amazon.com and AWS operational excellence, Amazon CodeGuru Reviewer, an ML-powered developer tool service, makes it easy to analyze code to ensure that security best practices hold everywhere. With Amazon CodeGuru Profiler, customers can identify expensive and inefficient methods in running applications and provide actionable steps to fix them, thereby saving money. Additionally, with Amazon DevOps Guru, customers can adopt an AIOps solution to improve an application's operational performance and availability. DevOps Guru identifies anomalous application behavior and surfaces critical issues that could cause potential outages or service disruptions and provides recommendations to remediate the issue.

"We are always looking for ways to reduce the amount of time our teams spend on resolving operational issues, and we are now using Amazon DevOps Guru and leveraging its ML-powered insights to help us identify, correlate, and remediate operational issues quickly. With the insights Amazon DevOps Guru provides, our teams can now quickly find issues without having to start from scratch trying to root cause problems. Our IT team has significantly reduced our mean time to recovery (MTTR), and they are saving hours upon hours of time resolving issues—all the while ensuring our customers have the best end-user experience possible."

—HCL Technologies

Continuous observability

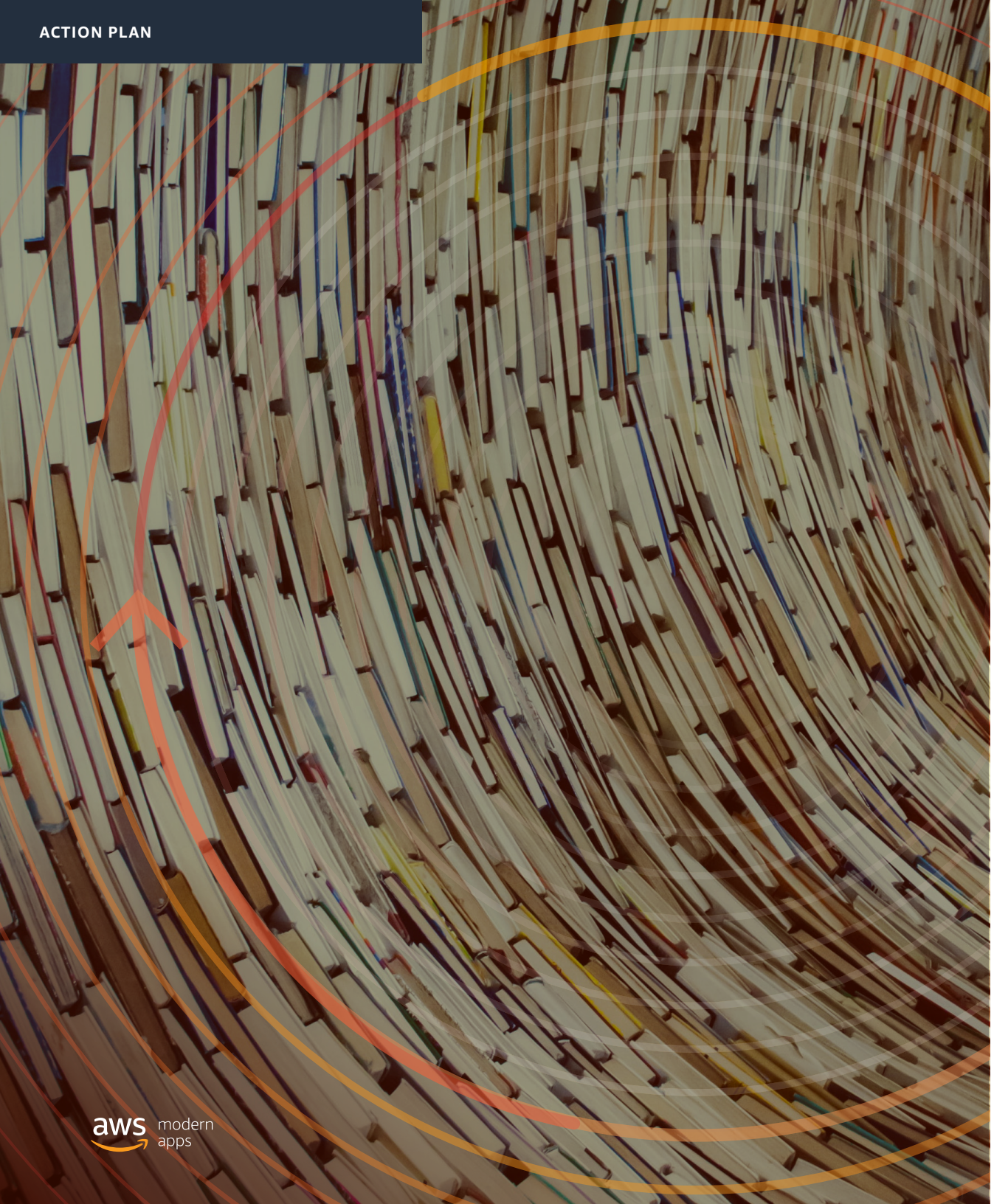
Continuous observability is the practice of being able to understand the status of your system at any given point in time. Observability enables you to detect, investigate, and remediate problems. Continuous observability gives engineering teams insight into their application health and performance so they can troubleshoot and resolve issues that need attention, ultimately improving their end users' experience. Cloud apps and resources generate billions of metrics, logs, and traces in a never-ending stream of data, and it can be challenging to analyze performance across distributed apps. Improve developer productivity with observability by identifying user impact from any source or finding broken or expensive code paths quickly.

The first step in continuous observability is to build observability dashboards with Amazon CloudWatch (or Amazon Managed Service for Prometheus, Amazon Managed Service for Grafana, or Amazon Distro for OpenTelemetry, based on preference). An observability dashboard addresses the challenge of staying on top of the activity in our cloud services. They are the human-facing views into our systems that provide concise summaries of how the system is behaving by displaying time-series metrics, logs, traces, and alarm data. After the dashboard has been built, customers should set CloudWatch alarms to continuously alert them to potential problems in their cloud environments.

"We use CloudWatch to create alerts for Amazon Simple Queue Service and for scaling actions for EC2 Auto Scaling Groups. These insights inform our scaling decisions, allowing us to get the most efficient usage out of our AWS compute resources without impacting our customers' experience. With CloudFormation, we can deploy logical stacks for our AWS resources used by our customer-facing applications, which provides us with repeatable and consistent deployments across AWS accounts. As a result, we have an easy way to compose application stacks using configuration as code rather than relying on error-prone manual configuration that can be inconsistent."

—Martin Costello, Senior Engineer, Just Eat

Fact:
AWS monitors an average of 1 quadrillion (1,000 trillion) metric observations each month



Build a dashboard for operational visibility

In this Amazon Builders' Library article, AWS Principal Engineer John O'Shea walks through how Amazon DevOps teams think about building dashboards to understand the state of their systems.

Learn more ›

Continuous compliance

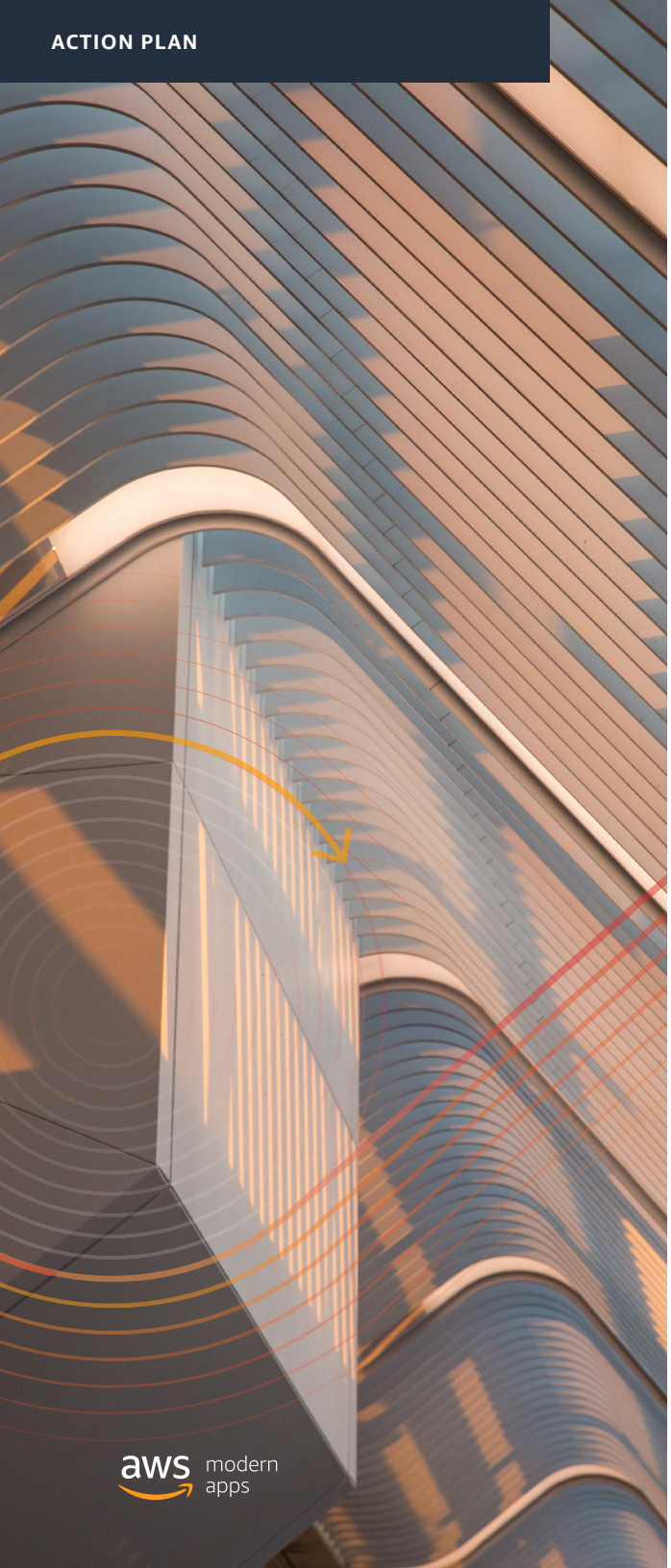
Compliance in the cloud is a whole new paradigm. The speed and scale of cloud resources make it impossible to use the traditional methods for compliance. Spreadsheets and static databases can't handle the resource churn. The key to success is to automate and simplify as much as possible. AWS provides a wide range of services to help customers streamline cloud compliance. One way to understand what services to use for compliance is to frame them with the Three Lines Model by The Institute of Internal Auditors, an internationally recognized authority on the practice of internal auditing. This model establishes that organizations manage risk through specific IT roles and responsibilities along three lines of defense. The first line manages risk, the second line oversees risk, and the third line provides assurance of risk.

AWS provides a wide range of services that fit well into each of these three lines. For example, for services that manage risk in the first line, we have AWS Config, AWS CloudTrail, and AWS Systems Manager. These services allow you to define and deploy individual controls to manage risk. In the second line, AWS Security Hub and Amazon CloudWatch help you oversee risk for your entire organization. And AWS Audit Manager is the perfect tool to provide assurance of risk, the third line, by allowing customers to automate the process of gathering evidence of compliance for audits.

But using these tools is only part of the story. Customers need to automate the process with continuous compliance. To adopt continuous compliance, we recommend leveraging the concept of compliance as code. Compliance as code allows you to define the elements of the Three Lines Model using tools like AWS CloudFormation. You can then test and deploy those elements using an automated pipeline, such as AWS CodePipeline. As new controls and evidence are needed, there is no need to manually implement them across each account. The code is updated in the control definition and pushed through the pipeline. The end result is a new control to manage risk that is immediately added as evidence for audit purposes. Continuous Compliance also has the added benefit of providing tangible assurance that you have the controls in place and that the collection of evidence has not been tampered with. Leveraging the Three Lines Model and automating the process with continuous compliance is the best approach for handling compliance at scale.

"With AWS, we've completely changed how we work and can now scale without adding headcount by automating compliance processes. Compliance can be complex to manage, but AWS Config helps even those of us with no deep knowledge in compliance to implement compliance with out-of-the-box templates. The dashboard provides users with information like noncompliant resources that can be used to improve security posture, and acts as a starting point for discussion with the rest of the organization of what we need to look at and remediate. As the digital group providing technical leadership through cloud technologies for both internal and external teams, it's important to have centralized visibility and control over our entire organization's security posture. The pre-packaged Config conformance pack templates have simplified this process and is a key factor in accelerating our overall migration to the cloud."

—Andrew Clark, Senior Solutions Architect, Baker Tilly Digital



Build a shared service platform

Every team and organization implements Dev+Ops a little bit differently. In many organizations, there is a central platform team that helps support development teams and reduce the operational burden by standardizing security, software delivery, monitoring, and networking. A shared services platform is a bespoke self-service interface for developers to use that is streamlined for deploying code. Central teams have control to define standards on security, software delivery, monitoring, and networking that must be used across all applications deployed. This allows developers to be more productive and gives platform teams more control.

AWS gives you all of the tools you need to build a shared service platform on AWS. For basic production use cases, AWS Copilot CLI makes it easy to create a "batteries included" development environment with multi-account CI/CD, security groups, and monitoring out of the box. For more complex situations, AWS provides a fully managed service called AWS Proton. AWS Proton is the first fully managed delivery service for container and serverless applications. Platform teams can use AWS Proton to connect and coordinate all the different tools needed for infrastructure provisioning, code deployments, monitoring, and updates. AWS Proton solves this by giving platform teams the tools they need to manage this complexity and enforce consistent standards while making it easy for developers to deploy their code using containers and serverless technologies. If your organization needs more control to build a shared service platform, many organizations have had success building with AWS CloudFormation and AWS Service Catalog to create and manage catalogs of IT services that are approved for use on AWS. Similarly, customers whose platform teams are comfortable with dynamic programming languages have used AWS CDK to build enablement platforms for their development teams.

"Our technologists can easily deploy our application builds to production using the AWS Service Catalog self-service portal, without having to open tickets. As a result, they can launch a new application stack in five minutes instead of the many weeks it would typically take. Because of AWS Service Catalog, we are enabling DevOps and automation at Wiley."[4]

—Meltem Dincer, Vice President of Platform Capabilities, Wiley

[4] *https://aws.amazon.com/solutions/case-studies/wiley/*

Regardless of where your organization is in your Dev+Ops journey, AWS offers a broad range of solutions designed to close the gap between developers and operations teams.

AWS Developer Tools are easy to get started with, cover major enterprise needs like security, compliance, high availability, and access control, and provide customers with proactive solutions that identify and prevent issues before they happen with advanced warnings and suggestions based on best practices. With AWS, customers can faster realize the benefits of the cloud and accelerate their innovation through our core set of Modern Dev+Ops practices, tools, and training.

Discover more about modern DevOps ›