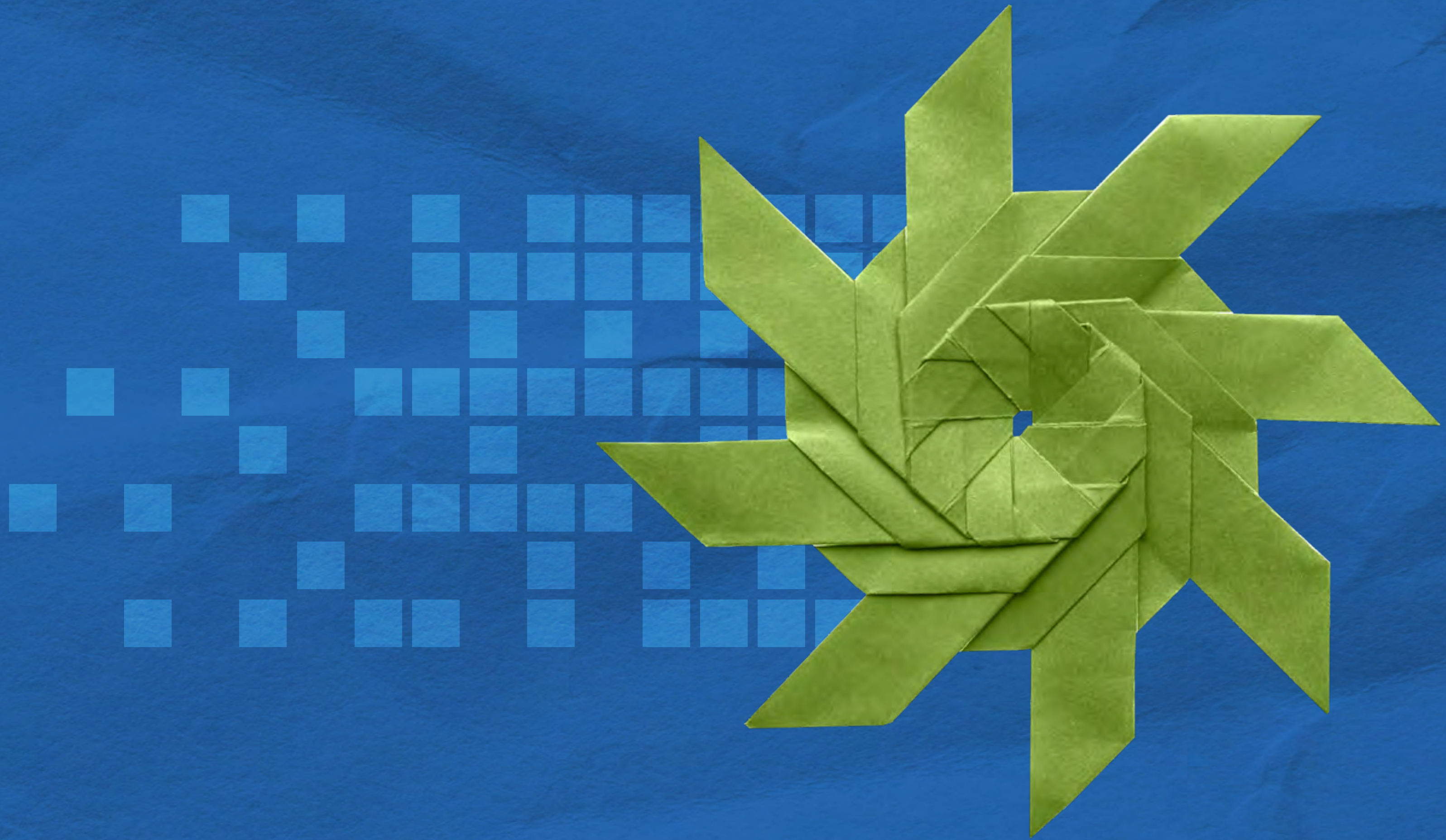
E-BOOK

FOUR GUIDELINES FOR DEVOPS SUCCESS

The role of monitoring in accelerating application development



NetApp

THE PACE OF DIGITAL TRANSFORMATION IS ACCELERATING, DRIVEN BY A NEW NORMAL WHERE ECONOMIC AND SOCIAL ACTIVITY ARE INCREASINGLY DIGITAL. FOR ORGANIZATIONS SEEKING TO GAIN OR MAINTAIN A COMPETITIVE ADVANTAGE, IT'S TIME TO TAKE A FRESH LOOK AT ADOPTING DEVOPS PRACTICES.

DevOps is an evolving philosophy and framework that encourages faster, better application development and faster release of software features or products. It encourages continuous communication, collaboration, integration, visibility, and transparency between application development teams (Dev) and their IT operations team (Ops) counterparts. A closer relationship between Dev and Ops permeates every phase of the DevOps lifecycle: from initial software planning to code, build, test, and release phases and on to deployment, operations, and ongoing monitoring.

There's no single path to DevOps. It isn't something you can buy or download to your phone. DevOps is about doing the work to transform your people, your processes, and your technology—breaking down silos, automating time-consuming procedures, and aligning around common goals.

This e-book highlights some essential guidelines in key areas of culture, process, and tooling to help you achieve DevOps success.

SOME COMMON TERMS

DevOps

According to Wikipedia, a set of practices that combines software development (Dev) and IT operations (Ops). DevOps aims to shorten the systems development lifecycle and provide continuous delivery with high software quality.

Agile

A set of project management practices that have evolved since the original Agile Manifesto was published in 2001. Agile practices emphasize self-organization and continuous adjustment to working practices.

Site Reliability Engineering (SRE)

A discipline that focuses on applying software engineering skills and practices to reduce the ongoing cost of operations.

Although Agile and SRE practices and closely correlated with high-functioning DevOps teams, neither is an essential element of DevOps.

GUIDELINE 1. CULTURE COMES FIRST

If your company isn’t ready for the transition to DevOps, success is far from assured. There are a few things that will help pave the way:

Understand your motivators. There are many possible reasons for deciding to move to DevOps: accelerate feature delivery, reduce software delivery costs, improve quality, or enable metric-driven decisions. Understanding your organization’s needs and the drivers behind your transition will help you establish clear goals.

Communicate your goals. After you determine your goals, you need to, communicate them clearly to all stakeholders across the company. The changes you make might be misunderstood or questioned. Changing the way development teams operate can come at a cost initially—in both capital and productivity—so it’s vital that everyone understands the big picture.

Manage change. A DevOps transition requires decisive adjustments to work practices and organizational structures. Significant management skills might be needed to keep everyone focused on goals and moving forward together.

Retrain versus hire. Your team might need new technical skills such as Git, Jenkins, Kubernetes, Ansible, and Terraform. Training existing staff on these technologies, rather than hiring new staff, is a good way to increase buy-in.

The changes that DevOps brings to roles, skillsets, team structure, and team relationships can be unsettling. Involve team members in decision-making and increase buy-in by discussing changes openly and highlighting opportunities for career development. Having key staff members as champions increases the velocity of both cultural and technical change.

GUIDELINE 2. TECHNOLOGY IS A MEANS, NOT AN END

DevOps is facilitated by tooling, not defined by it. A common misstep is a belief that new tools will change habits, behaviors, and development practices all by themselves. Improving communication and increasing levels of automation should be one of your top objectives, no matter what tooling you decide on.

Choose the tools that you need. Don’t simply adopt new technologies because they are trendy. Keep in mind the context in which development work is done at your company. Pay attention to what other organizations are doing, but be mindful that the tooling that worked for one company might not be ideal for your needs. There is no single “right” set of DevOps tools.

Take advantage of existing strengths. You’ll have more success if you play to your team’s strengths when choosing languages, platforms, and other tools. For example, if an application will run equally well in virtual machines (VMs) or containers, but you currently lack containers and Kubernetes skills, VMs might be a smarter choice.

Invest in training. Make sure that your team gets trained to use new processes and tooling effectively. Don’t just expect them to pick up the skills on their own.

Focus on continuous improvement. Rather than changing everything at once, introduce changes gradually. Consider which processes and tools will deliver the biggest immediate benefits, and introduce those changes first to help build momentum.

Choose team-wide tools early. DevOps teams typically have both Dev-focused and Ops-focused members, with some separate Dev and Ops tools. However, there are tools that everyone will use, such as communication, collaboration, bug-reporting, monitoring, and other tools. Putting these tools in place quickly will foster a more cohesive team and encourage shared work practices.

GUIDELINE 3. DON'T FORGET THE "OPS" IN DEVOPS

The cloud (private and public) has made it easier to deploy infrastructure for development and production but harder to control costs and optimize use. As part of your DevOps effort, your team will almost certainly use cloud resources to speed time to market. Typically, Dev teams are better at consuming resources than they are at releasing them. Therefore, tracking and controlling resource consumption and identifying waste have now become critical Ops functions that you ignore at your peril.

Technologies such as microservices and distributed tracing require new approaches to monitoring. You can't afford to sacrifice traditional Ops virtues—budget discipline, system and application availability, security and compliance—for velocity. Unless you evolve your tooling, you might expose DevOps projects to increased risk of failure and runaway costs that undermine your efforts.

TO BE CONTINUOUS

DevOps embraces continuous improvement and automation. Common practices include:

Continuous development. Spans the planning and coding phases of the DevOps lifecycle. Version-control mechanisms might be involved.

Continuous testing. Incorporates automated, prescheduled, continuous code tests as application code is written or updated, speeding the delivery of code to production.

Continuous integration (CI). Bring configuration management tools together with tools that track how much of the code being developed is ready for production. CI requires rapid feedback between testing and development to identify and resolve code issues quickly.

Continuous delivery. Automates the delivery of code changes, after testing, to a preproduction or staging environment. A staff member might then decided to promote such code changes into production. A company doing continuous deployment might release code or feature changes several times per day.

Continuous monitoring. Ongoing monitoring of both the code in operation and the underlying infrastructure that supports it. Continuous monitoring involves a feedback loop that reports on bugs or issues and then makes its way back to development.

GUIDELINE 4. MAKE MONITORING EVERYONE’S JOB

Monitoring is a skill that everyone on your DevOps team needs. It’s not something you simply assign to someone—or to a team of people—and forget about. No one can design effective monitoring for applications or infrastructure if they aren’t directly connected with it.

As your organization moves to DevOps, you’ll need to identify the service-level indicators (SLIs) that confirm your systems are meeting SLAs. You might have to manage hybrid applications that rely on services in multiple clouds, such as an application with customer-facing components in the cloud coupled with an on-premises database.

Monitoring tools need to satisfy several criteria. They need to:

- Enable monitoring in both private and public clouds
- Satisfy Dev needs by reporting application or service-specific SLIs
- Satisfy Ops needs by reporting infrastructure-level SLIs
- Bridge between Dev and Ops SLIs and facilitate cross-domain troubleshooting

One of the goals of DevOps is to bring Dev and Ops together, allowing them to understand and participate in each other’s work. Intelligent tooling helps facilitate this transition.

NetApp

GETTING STARTED WITH DEVOPS

Moving to DevOps is a process of continuous improvement; don't expect to learn everything you'll need to know upfront. Make iterative improvements, and publicize your successes to keep up momentum.

To get started with DevOps, you need monitoring tools that improve operational efficiency and performance. NetApp offers a cloud-based tool that can reduce troubleshooting time, accurately predict performance needs, and help control costs: NetApp Cloud Insights. It's a software-as-a-service (SaaS) monitoring tool that gives you actionable knowledge of your infrastructure.

DevOps Resources

- Why DevOps, and Why Now?
- What is DevOps
- ThePub: DevOps

Cloud Insights Resources

- NetApp Cloud Insights: A New Way to Monitor Your Cloud Infrastructure
- Enhance Monitoring by Understanding the Relationships Between Resources
- Keep Cloud Costs at Bay with Queries in Cloud Insights (blog)

If you're ready to learn more, NetApp has various DevOps resources targeted to different needs.

To see Cloud Insights in action for yourself, register for a 30-day free trial. Visit NetApp Cloud Central to learn more about NetApp Cloud Insights and start your free trial.

ABOUT NETAPP

In a world full of generalists, NetApp is a specialist. We're focused on one thing, helping your business get the most out of your data. NetApp brings the enterprise-grade data services you rely on into the cloud, and the simple flexibility of cloud into the data center. Our industry-leading solutions work across diverse customer environments and the world's biggest public clouds. As a cloud-led, data-centric software company, only NetApp can help build your unique data fabric, simplify and connect your cloud, and securely deliver the right data, services and applications to the right people—anytime, anywhere.

Learn more at www.netapp.com.