New Relic®

eBook

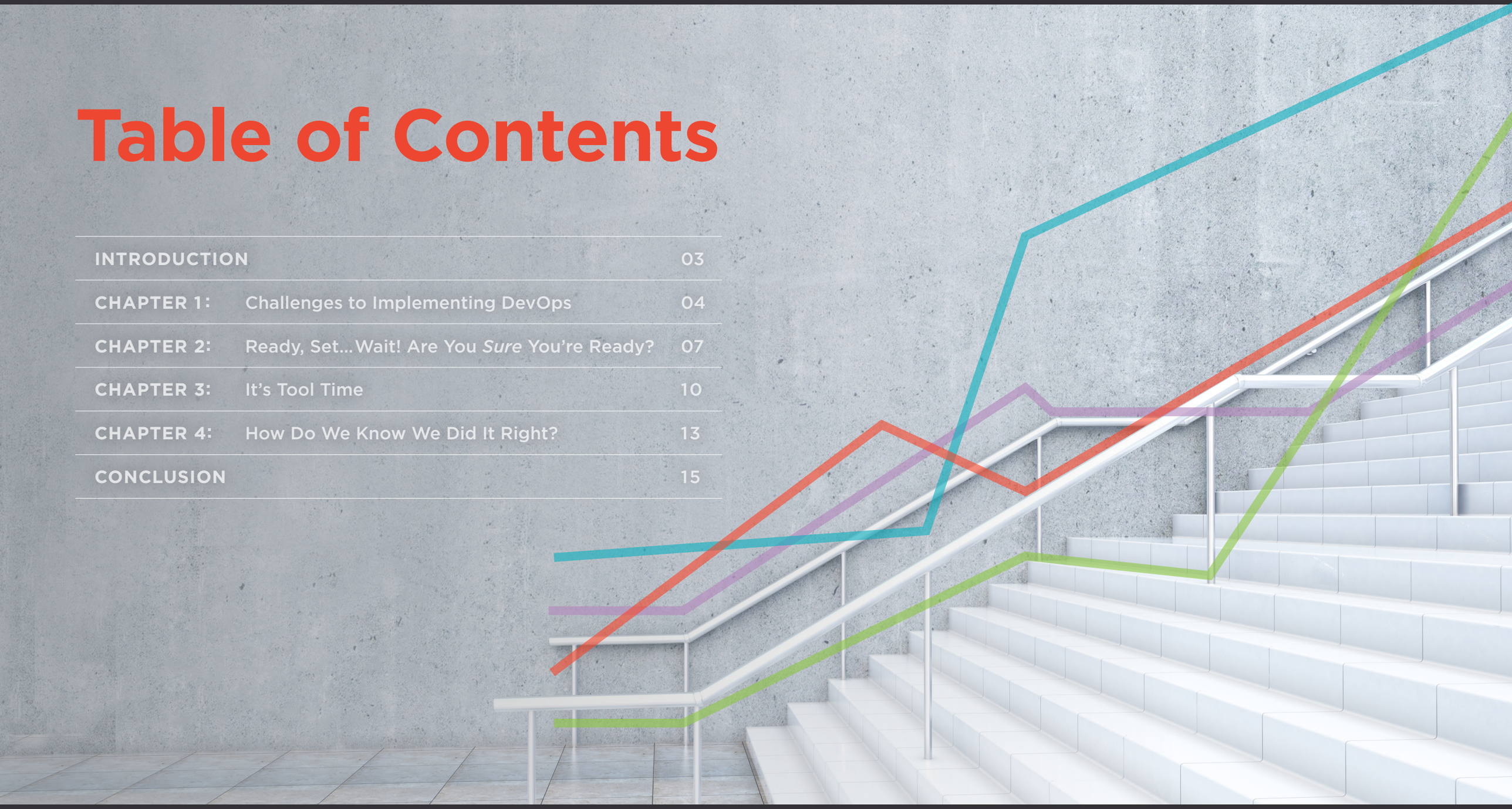
Kickstarting DevOps

How to get everyone on board and successfully make the transition

© 2017 New Relic, Inc | US +888-643-8776 | www.newrelic.com | www.twitter.com/newrelic | blog.newrelic.com

Table of Contents

INTRODUCTION		03
CHAPTER 1:	Challenges to Implementing DevOps	04
CHAPTER 2:	Ready, Set...Wait! Are You *Sure* You're Ready?	07
CHAPTER 3:	It's Tool Time	10
CHAPTER 4:	How Do We Know We Did It Right?	13
CONCLUSION		15



© 2017 New Relic, Inc | US +888-643-8776 | www.newrelic.com | www.twitter.com/newrelic | blog.newrelic.com

New Relic.

Introduction

With nearly three-quarters of organizations now using DevOps[1], it's probably an easy decision to adopt DevOps practices in your business. After all, to keep up with the rapid pace of business today, you need DevOps to help you accelerate deployments while improving software quality and business outcomes.

However, the rest of your organization may not be so enthusiastic. This eBook will help you familiarize yourself with the common barriers to DevOps adoption so that you can win over the skeptics—and get your organization's DevOps movement rolling.

[1] "RightScale 2016: State of the Cloud Report: DevOps Trends," RightScale, May 2016.

© 2017 New Relic, Inc | US +888-643-8776 | www.newrelic.com | www.twitter.com/newrelic | blog.newrelic.com

CHAPTER 1

Challenges to Implementing DevOps

Challenges to Implementing DevOps

Many people fear change and, without a doubt, DevOps represents change. However, much of the uncertainty and hesitation that is attached to the notion of DevOps can be lessened simply by educating individual stakeholders on how DevOps will affect them, its pluses as well as minuses, and, most important, their role in shaping your organization's approach to DevOps.

Convincing Developers? Easy. Operations? Not So Much.

Developers are often the easiest converts to DevOps for several reasons. First, they're more accustomed to change than those from other disciplines. Unless they graduated from engineering school yesterday, most software engineers are coding today in a language that they learned in the workplace, not a university classroom. In addition, the buzz about DevOps often emphasizes the quality of life and job satisfaction benefits to developers, including faster provisioning, and thus greater room for creativity and innovation. And let's face it: developers just like new toys, and DevOps comes with lots of them.

Operations professionals react differently. Many fear that with DevOps, developers will dump untested, failure-prone software into production, destabilizing the system. While there's ample evidence that the reverse is usually true, keeping the system running is the number one imperative for operations, so DevOps can be a harder sell for them. The trick here is to help operations personnel understand that faster deployment cycles, with significantly fewer changes per cycle, mean that problems are much easier to identify and rollbacks should be painless.

Will DevOps Lead to NoOps?

Another concern for operations is that their function will go away, a fear that is exacerbated by the talk about the NoOps (No Operations) movement. The idea of NoOps is that automation, abstraction, and outsourcing to cloud service providers eliminates the need for IT operations entirely.[2]

While the idea has been around for years, new technologies such as serverless computing are paving the way for complete automation of the production operation process.

[2] Beth Pariseau, "Serverless computing supporters ponder NoOps," TechTarget, http://searchitoperations.techtarget.com/news/450297503/Serverless-computing-supporters-ponder-NoOps, May 31, 2016.

© 2017 New Relic, Inc | US +888-643-8776 | www.newrelic.com | www.twitter.com/newrelic | blog.newrelic.com

Operations may suspect this is one step on the path to being eliminated altogether. However, as serverless automation evolves, it just means that operations will be focusing further upstream than before. Ops folks "will still need to ensure that code is packaged and provisioned correctly… the new operations professional will apply systems thinking to holistically look at how business applications and customer experience can be improved."[3]

Everybody Gets into the Act

DevOps impacts people in the organization in a number of areas. It requires new skills; for example, system administrators may have to develop new tools, a skill they may not have learned or used often. The pace of DevOps can be challenging, especially for developers who work in the more linear world of waterfall development. Line managers may resist the formation of cross-functional teams based on concerns about the difficulty of managing in such an environment—in fact, this objection is a major inhibitor to DevOps adoption in larger enterprises with hierarchical management systems. Executives fret that they will not be able to attract and retain talent with the new skills and abilities necessary for DevOps and worry about undermining salary guidelines to compete for scarce software professionals.

Have You Thought About…?

DevOps changes software workflows and that has implications as well. For one thing, legacy linear management tools are often not suited for DevOps, requiring capital investment in new tools (although many organizations are turning to open source and home-grown tools as a way to mitigate the expense). DevOps also requires much more cross-functional collaboration, which means that teams need to be looking at the same source of data to agree on priorities and measure the progress of their efforts (see more about KPIs on p. 9).

In addition, the intertwined workflows are harder to visualize and document, making regulatory compliance more difficult and time-consuming. Security teams need to adjust as well:

> *"[DevOps] represents a change in mindset for security teams because 'fast and easy' is not something that comes naturally to them."*[4]

Security and other teams do not need to be left out in the cold. A good DevOps implementation will ensure that all teams' needs are met. But to get there, you need to make sure that everyone adjusts their traditional workflows and views to be more fluid and based on quicker iteration cycles.

[3] Beth Pariseau, "Serverless computing supporters ponder NoOps," TechTarget, http://searchitoperations.techtarget.com/news/450297503/Serverless-computing-supporters-ponder-NoOps, May 31, 2016.

[4] Warwick Ashford, "DevOps a black hole for security," ComputerWeekly.com, http://www.computerweekly.com/news/450300208/DevOps-a-black-hole-for-security, July 13, 2016.

© 2017 New Relic, Inc | US +888-643-8776 | www.newrelic.com | www.twitter.com/newrelic | blog.newrelic.com

CHAPTER 2

Ready, Set...Wait! Are You *Sure* You're Ready?

Ready, Set...Wait! Are You *Sure* You're Ready?

Let's say you're an IT manager who wants to introduce DevOps into your organization. You've seen how DevOps is transforming IT in a wide range of companies and helping IT get new products to market faster, and you want to do the same in your company. So what do you do now? You can start by following these four steps.

Step 1: Align With Business Goals

You're probably chomping at the bit to dive into the process, but before you do, put on your business hat and go talk to your business counterparts. How many? As many as you can. The more clearly you understand the needs and wants of the business side, the better you can leverage DevOps to meet those needs. It's better if you don't even mention DevOps, because they may not know what it is or may have preconceived notions about it. So as you start these conversations, try asking questions along these lines:

- How do you measure success?
- What are your goals and objectives for the next year?
- Which of those goals are most at risk?
- What is holding you back from blowing the doors off?

Step 2: Characterize Your Existing Environment

As obvious as it may seem to define the starting point, many organizations simply plunge right in—and then have no way to adequately assess their progress. Invest the time to characterize your current state as thoroughly as possible. Use discovery tools and processes to develop a clear inventory of all devices in the environment. Document the configurations of every configurable component in the infrastructure. Record any information that might be required to reconstruct the starting point. You will never actually do so, but capturing all the knowledge required to return to the starting point means that you have all the information necessary to monitor the changes as your DevOps initiative progresses. This capability will become invaluable for identifying drift from your known, trusted state.

Step 3: Get Buy-in From All Stakeholders

Much of the DevOps discussion focuses on development and operations, but the implications of DevOps extend throughout the organization. Therefore it is vital to clearly define the processes and identify the owners who will be impacted, being sure to include them in the planning stage. Establish open channels of communication. Be transparent

© 2017 New Relic, Inc | US +888-643-8776 | www.newrelic.com | www.twitter.com/newrelic | blog.newrelic.com

by keeping everyone in the loop as the project progresses—particularly when you hit snags. By securing the buy-in of every stakeholder, you increase the probability of success.

Step 4: Establish and Track Key Performance Indicators

Because DevOps fundamentally changes the way that IT does its job, having the right metrics is critical to evaluating the success of a DevOps initiative. Key performance indicators (KPIs) act as a "canary in the coal mine," providing an early warning when the system begins to degrade. An initial set of KPIs might include:[5]

- **Deployment frequency:** Companies with DevOps cultures deploy much more frequently. In fact, according to a survey from Puppet and DORA, high-performing organizations deploy 200 times more frequently than low performers, with an average of 1,460 deploys per year.
- **Change lead time:** Making changes quickly is the very definition of agility. High-performing DevOps teams average less than one hour of lead time for changes, while low performers take between one to six months.
- **Mean time to recover (MTTR):** Every organization has failures, but DevOps companies recover in minutes, not hours. Having precise measurements of MTTR helps IT managers monitor the people, processes, and technology that enable rapid recovery and head off problems before they result in significant downtime.
- **Change fail rate:** Even with rapid recovery, it's better not to fail at all. High-performing DevOps groups have a 0–15% change failure rate. Any spike in failures allows IT managers to find and fix the organization problem that is driving the increase and ensure that change fail rates remain low.

Now that you have a better idea of how to get started, let's take a deeper dive into the specific tools you'll likely use in a DevOps environment.

[5] "2016 State of DevOps Report," Puppet Inc. and DORA, https://puppet.com/resources/whitepaper/2016-state-of-devops-report, June 2016.

© 2017 New Relic, Inc | US +888-643-8776 | www.newrelic.com | www.twitter.com/newrelic | blog.newrelic.com

CHAPTER 3

It's Tool Time

It's Tool Time

While having the right tools won't magically transform your organization into a lean, mean DevOps machine, it's hard to overstate the importance of tools in a DevOps environment. If DevOps is in your future, then understanding the tools landscape is essential.

We'll be up front about this: We can't advise you about the definitive set of tools for your DevOps environment (however, we can make a pretty strong case for using New Relic to measure and enable your DevOps success). Every enterprise is different, every IT group is different, every environment is different, and the tools are changing so fast that anything you read about tools is nearly obsolete by the time you read it.

The goal of this section is simply to give you a head start in the tool arena by defining the basic categories and listing some of the favorite tools used by New Relic's engineering team. And if you find yourself wanting an even deeper discussion of a few specific tools, check out *The 5 Unsung Tools of DevOps*, an eBook written by New Relic Site Reliability engineer Jonathan Thurman.

Configuration Management

Configuration management—sometimes called infrastructure automation—refers to tracking and controlling changes to the software code base. When your app crashes or some other problem arises, configuration management helps determine what was changed and who changed it.

It's an essential practice for establishing and keeping consistent product performance, especially when various developers and system administrators are working on the same code base, as is the case in DevOps environments. Some of the more commonly used configuration management tools include:

- Ansible
- CFEngine
- Chef
- Puppet
- SaltStack
- Ubuntu Juju

Test-and-Build Systems

Test-and-build systems automate a number of developer tasks such as compiling source code into binary executables, running texts, and creating documentation. Typical open source tools for building and testing code include:

- Ant
- Gradle
- Jenkins
- Maven

© 2017 New Relic, Inc | US +888-643-8776 | www.newrelic.com | www.twitter.com/newrelic | blog.newrelic.com

Application Deployment

Also known as release automation, application deployment tools are critical to continuous delivery of software—one of the key tenets of DevOps. The most popular standalone application deployment tool for Ruby-based Web applications is Capistrano, a remote server automation tool. Bamboo, Fabric, Jenkins, Puppet, and Team Foundation Server (TFS) are also popular tools for automating application deployment.

Monitoring Tools

There are two monitoring needs in DevOps. Application performance monitoring (APM) provides code-level visibility that enables quick identification of performance issues as well as rapid remediation. The more fully featured APM tools provide trending reports and alerts.

Server monitoring operates at the infrastructure level, allowing reliability engineers to track server health in cloud, physical, and hybrid environments. These tools show capacity, memory, and CPU status for each server so that problems can be addressed early—ideally, before they impact application performance.

Containers

Containers are self-contained execution environments. They are very lightweight virtualization solutions for the operating system versus virtualization that refers to hardware. Containers have their own isolated CPU, memory, block I/O, and network resources that share the kernel of the host operating system.

> *"DevOps teams can benefit from implementing containers into their infrastructure and pipeline, as a solid container-based development workflow not only allows for flexible and agile application development but with testing, staging, and bug fixes. Containers have seen a variety of improvements in the last year including better container visibility, enhanced log management, improved security, and a focus on scalability."*[6]

While Docker is the leader in the container market, you can also choose from Kubernetes or Mesos based on the scale, infrastructure, and availability needs of your organization.

Collaboration and Communication

For DevOps teams, collaboration and communication are essential. That's why you should look at finding a tool that helps bring your teams together to communicate effectively, share work, and collaborate. Here are few popular tools on the market today:

- Confluence
- Guru
- HipChat
- Invision
- Slack
- Spark
- Trello

[6] CK Oliver, "Demystifying Containers for a Better DevOps Experience," Forbes, http://www.forbes.com/sites/ckoliver/2016/04/29/demystifying-containers-for-a-better-devops-experience/#13252fff109c, April 29, 2016.

© 2017 New Relic, Inc | US +888-643-8776 | www.newrelic.com | www.twitter.com/newrelic | blog.newrelic.com

CHAPTER 4

How Do We Know We Did It Right?

How Do We Know We Did It Right?

The proof is in the pudding: You accelerated software delivery. You have fewer defects in the software, faster resolution of problems, and better allocation of your limited resources. Pat yourself on the back—you achieved your goals.

But what does the business think of your outcomes? Have you demonstrated that what you do has a positive impact on the business? To do this, you must link and balance the goals of faster (speed of delivery), better software (high-performing, quality software that delivers a good customer experience) to goals for innovation and business success.

For instance, agreeing on the business goals for your software helps everyone on the DevOps team relate what they're doing back to a measurable set of indicators of success. This data-driven approach improves prioritization and decision making—from which features to include in your software to resource allocation and the size of your DevOps team.

Read more about how to measure DevOps success in our ebook, "DevOps Without Measurement Is a Fail."

© 2017 New Relic, Inc | US +888-643-8776 | www.newrelic.com | www.twitter.com/newrelic | blog.newrelic.com

Conclusion

The gap is growing between companies that "get" DevOps and those that don't.

High-performing organizations deploy 200 times more frequently, have 2,555 times faster lead times, and recover 24 times faster from failed changes, with three times lower change fail rates than low-performing companies.[7]

To become a high-performing organization, you need a strong DevOps program. That will mean overcoming a number of obstacles and getting off to the right start.

One of the keys to DevOps success lies in taking advantage of the right tools, many of which are open source. Automating tasks such as version control, server configuration, testing, and even the deployment process itself pay huge benefits.

You also need detailed, real-time, and historical data to track and monitor your speed, application performance, customer experience, and business success. Using a single digital intelligence platform like New Relic simplifies and streamlines the management of your DevOps efforts. To learn more about DevOps, visit: http://newrelic.com/devops.

About New Relic

New Relic is a leading digital intelligence company, delivering full-stack visibility and analytics with more than 14,000 paid business accounts. The New Relic Digital Intelligence Platform provides actionable insights to drive digital business results. Companies of all sizes trust New Relic to monitor application and infrastructure performance so they can quickly resolve issues and improve digital customer experiences. Learn more at newrelic.com.

[7] "2016 State of DevOps Report," Puppet Inc. and DORA, https://puppet.com/resources/whitepaper/2016-state-of-devops-report, June 2016.

© 2017 New Relic, Inc | US +888-643-8776 | www.newrelic.com | www.twitter.com/newrelic | blog.newrelic.com

©2008-17 New Relic, Inc. All rights reserved.