



/THEORY/IN/PRACTICE

# Web Operations

Keeping the Data on Time

(影印版)

O'REILLY®

東南大學出版社

John Allspaw  
& Jesse Robbins 編

---

网站运维 (影印版)

Web Operations

*John Allspaw, Jesse Robbins* 编

O'REILLY®

*Beijing • Cambridge • Farnham • Köln • Sebastopol • Taipei • Tokyo*

O'Reilly Media, Inc. 授权东南大学出版社出版

东南大学出版社

## 图书在版编目 (CIP) 数据

网站运维: 英文 / (美) 阿尔斯帕瓦 (Allspaw, J.),  
(美) 罗宾斯 (Robbins, J.) 著. — 影印本. — 南京: 东南大学出版社, 2011.1

书名原文: Web Operations

ISBN 978-7-5641-2502-8

I. ①网… II. ①阿… ②罗… III. ①网站—开发  
IV. ①TP393.092

中国版本图书馆 CIP 数据核字 (2010) 第 212892 号

江苏省版权局著作权合同登记

图字: 10-2010-289 号

©2010 by O'Reilly Media, Inc.

Reprint of the English Edition, jointly published by O'Reilly Media, Inc. and Southeast University Press, 2010. Authorized reprint of the original English edition, 2010 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由 O'Reilly Media, Inc. 出版 2010。

英文影印版由东南大学出版社出版 2010。此影印版的出版和销售得到出版权和销售权的所有者——O'Reilly Media, Inc. 的许可。

版权所有, 未得书面许可, 本书的任何部分和全部不得以任何形式重制。

## 网站运维 (影印版)

---

出版发行: 东南大学出版社

地 址: 南京四牌楼 2 号 邮编: 210096

出 版 人: 江建中

网 址: <http://www.seupress.com>

电子邮件: [press@seu.edu.cn](mailto:press@seu.edu.cn)

印 刷: 扬中市印刷有限公司

开 本: 787 毫米 × 980 毫米 16 开本

印 张: 21 印张

字 数: 411 千字

版 次: 2011 年 1 月第 1 版

印 次: 2011 年 1 月第 1 次印刷

书 号: ISBN 978-7-5641-2502-8

印 数: 1~1700 册

定 价: 54.00 元 (册)

---

本社图书若有印装质量问题, 请直接与读者服务部联系。电话 (传真): 025-83792328

# Foreword

## **IT'S BEEN OVER A DECADE SINCE THE FIRST WEBSITES REACHED REAL SCALE.**

We were there then, in those early days, watching our sites growing faster than anyone had seen before or knew how to manage. It was up to us figure out how to keep everything running, to make things happen, to get things done.

While everyone else was at the launch party, we were deep in the bowels of the data-center racking and stacking the last servers. Then we sat at our desks late into the night, our faces lit with the glow of logfiles and graphs streaming by.

Our experiences were universal: Our software crashed or couldn't scale. The databases crashed and data was corrupted, while every server, disk, and switch failed in ways the manufacturer absolutely, positively said it wouldn't. Hackers attacked—first for fun and then for profit. And just when we got things working again, a new feature would be pushed out, traffic would spike, and everything would break all over again.

In the early days, we used what we could find because we had no budget. Then we grew from mismatched, scavenged machines hidden in closets to megawatt-scale datacenters spanning the globe filled with the cheapest machines we could find.

As we got to scale, we had to deal with the real world and its many dangers. Our datacenters caught fire, flooded, or were ripped apart by hurricanes. Our power failed. Generators didn't kick in—or started and then ran out of fuel—or were taken down when someone hit the Emergency Power Off. Cooling failed. Sprinklers leaked. Fiber was cut by backhoes and squirrels and strange creatures crawling along the seafloor.

Man, machine, and Mother Nature challenged us in every way imaginable and then surprised us in ways we never expected.

We worked from the instant our pagers woke us up or when a friend innocently inquired, “Is the site down?” or when the CEO called scared and furious. We were always the first ones to know it was down and the last to leave when it was back up again.

Always.

Every day we got a little smarter, a little wiser, and learned a few more tricks. The scripts we wrote a decade ago have matured into tools and languages of their own, and whole industries have emerged around what we do. The knowledge, experiences, tools, and processes are growing into an art we call Web Operations.

We say that Web Operations is an art, not a science, for a reason. There are no standards, certifications, or formal schooling (at least not yet). What we do takes a long time to learn and longer to master, and everyone at every skill level must find his or her own style. There’s no “right way,” only what works (for now) and a commitment to doing it even better next time.

The Web is changing the way we live and touches every person alive. As more and more people depend on the Web, they depend on us.

Web Operations is work that matters.

—Jesse Robbins

*The contributors to this book have donated their payments to the 826 Foundation, which helps kids learn to love reading at places like the Superhero Supply Company, the Greenwood Space Travel Supply Company, and the Liberty Street Robot Supply & Repair Shop.*

# Preface

**DESIGNING, BUILDING, AND MAINTAINING A GROWING WEBSITE** has unique challenges when it comes to the fields of systems administration and software development. For one, the Web never sleeps. Because websites are globally used, there is no “good” time for changes, upgrades, or maintenance windows, only fewer “bad” times. This also means that outages are guaranteed to affect someone, somewhere using the site, no matter what time it is.

As web applications become an increasing part of our daily lives, they are also becoming more complex. With that complexity comes more parts to build and maintain and, unfortunately, more parts to fail. On top of that, there are requirements for being fast, secure, and always available across the planet. All these things add up to what’s become a specialized field of engineering: web operations.

This book was conceived to gather insights into this still-evolving field from web veterans around the industry. Jesse Robbins and I came up with a list of tip-of-iceberg topics and asked these experts for their hard-earned advice and stories from the trenches.

## How This Book Is Organized

The chapters in this book are organized as follows:

Chapter 1, *Web Operations: The Career* by Theo Schlossnagle, describes what this field actually encompasses and underscores how the skills needed are gained by experience and less about formal education.

Chapter 2, *How Picnik Uses Cloud Computing: Lessons Learned* by Justin Huff, explains how Picnik.com went about deploying and sustaining its infrastructure on a mix of on-premise hardware and cloud services.

Chapter 3, *Infrastructure and Application Metrics* by Matt Massie and myself, discusses the importance of gathering metrics from both your application and your infrastructure, and considerations on how to gather them.

Chapter 4, *Continuous Deployment* by Eric Ries, gives his take on the advantages of deploying code to production in small batches, frequently.

Chapter 5, *Infrastructure as Code* by Adam Jacob, gives an overview about the theory and approaches for configuration and deployment management.

Chapter 6, *Monitoring* by Patrick Debois, discusses the various considerations when designing a monitoring system.

Chapter 7, *How Complex Systems Fail*, is Dr. Richard Cook's whitepaper on systems failure and the nature of complexity that is often found in web architectures. He also adds some web operations-specific notes to his original paper.

Chapter 8, *Community Management and Web Operations*, is my interview with Heather Champ on the topic of how outages and degradations should be handled on the human side of things.

Chapter 9, *Dealing with Unexpected Traffic Spikes* by Brian Moon, talks about the experiences with huge traffic deluges at Dealnews.com and what they did to mitigate disaster.

Chapter 10, *Dev and Ops Collaboration and Cooperation* by Paul Hammond, lists some of the places where development and operations can come together to enable the business, both technically and culturally.

Chapter 11, *How Your Visitors Feel: User-Facing Metrics* by Alistair Croll and Sean Power, discusses metrics that can be used to illustrate what the real experience of your site is.

Chapter 12, *Relational Database Strategy and Tactics for the Web* by Baron Schwartz, lays out common approaches to database architectures and some pitfalls that come with increasing scale.

Chapter 13, *How to Make Failure Beautiful: The Art and Science of Postmortems* by Jake Loomis, goes into what makes or breaks a good postmortem and root cause analysis process.

Chapter 14, *Storage* by Anoop Nagwani, explores the gamut of approaches and considerations when designing and maintaining storage for a growing web application.

Chapter 15, *Nonrelational Databases* by Eric Florenzano, lists considerations and advantages of using a growing number of "nonrelational" database technologies.

Chapter 16, *Agile Infrastructure* by Andrew Clay Shafer, discusses the human and process sides of operations, and how agile philosophy and methods map (or not) to the operational space.

Chapter 17, *Things That Go Bump in the Night (and How to Sleep Through Them)* by Mike Christian, takes you through the various levels of availability and Business Continuity Planning (BCP) approaches and dangers.

## Who This Book Is For

This book is for developers; systems administrators; and database, network, or any other engineer who is tasked with operating a web application. The topics covered here are all applicable to the field of web operations, which is a continually evolving field.

## Conventions Used in This Book

The following typographical conventions are used in this book:

### *Italic*

Indicates new terms, URLs, email addresses, filenames, and file extensions.

### Constant width

Used for program listings, as well as within paragraphs to refer to program elements such as variable or function names, databases, data types, environment variables, statements, and keywords.

### **Constant width bold**

Shows commands or other text that should be typed literally by the user.

### *Constant width italic*

Shows text that should be replaced with user-supplied values or by values determined by context.

## Using Code Examples

This book is here to help you get your job done. In general, you may use the code in this book in your programs and documentation. You do not need to contact us for permission unless you're reproducing a significant portion of the code. For example, writing a program that uses several chunks of code from this book does not require permission. Selling or distributing a CD-ROM of examples from O'Reilly books does require permission. Answering a question by citing this book and quoting example code does not require permission. Incorporating a significant amount of example code from this book into your product's documentation does require permission.

We appreciate, but do not require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example: "*Web Operations: Keeping Data On Time*, edited by John Allspaw and Jesse Robbins. Copyright 2010 O'Reilly Media, Inc., 978-1-449-37744-1."

If you feel your use of code examples falls outside fair use or the permission given here, feel free to contact us at [permissions@oreilly.com](mailto:permissions@oreilly.com).



## How to Contact Us

Please address comments and questions concerning this book to the publisher:

O'Reilly Media, Inc.  
1005 Gravenstein Highway North  
Sebastopol, CA 95472  
800-998-9938 (in the United States or Canada)  
707-829-0515 (international or local)  
707-829-0104 (fax)

We have a web page for this book, where we list errata, examples, and any additional information. You can access this page at:

*<http://oreilly.com/catalog/9781449377441>*

To comment or ask technical questions about this book, send email to:

*[bookquestions@oreilly.com](mailto:bookquestions@oreilly.com)*

For more information about our books, conferences, Resource Centers, and the O'Reilly Network, see our website at:

*<http://oreilly.com>*

## Safari® Books Online

**Safari<sup>®</sup>** Safari Books Online is an on-demand digital library that lets you easily search over 7,500 technology and creative reference books and videos to find the answers you need quickly.

With a subscription, you can read any page and watch any video from our library online. Read books on your cell phone and mobile devices. Access new titles before they are available for print, and get exclusive access to manuscripts in development and post feedback for the authors. Copy and paste code samples, organize your favorites, download chapters, bookmark key sections, create notes, print out pages, and benefit from tons of other time-saving features.

O'Reilly Media has uploaded this book to the Safari Books Online service. To have full digital access to this book and others on similar topics from O'Reilly and other publishers, sign up for free at *<http://my.safaribooksonline.com>*.

## Acknowledgments

John Allspaw would like to thank Elizabeth, Sadie, and Jack for being very patient while I worked on this book. I'd also like to thank the contributors for meeting their deadlines on a tight schedule. They all of course have day jobs.

Jesse Robbins would like to thank John Allspaw for doing the majority of the work in creating this book. It would never have happened without him.

# CONTENTS

	<b>Foreword</b> . . . . .	<b>xi</b>
	<b>Preface</b> . . . . .	<b>xiii</b>
<b>1</b>	<b>Web Operations: The Career</b> . . . . .	<b>1</b>
	<i>Theo Schlossnagle</i>	
	Why Does Web Operations Have It Tough?	2
	From Apprentice to Master	4
	Conclusion	9
<b>2</b>	<b>How Picnik Uses Cloud Computing: Lessons Learned</b> .	<b>11</b>
	<i>Justin Huff</i>	
	Where the Cloud Fits (and Why!)	12
	Where the Cloud Doesn't Fit (for Picnik)	20
	Conclusion	20
<b>3</b>	<b>Infrastructure and Application Metrics</b> . . . . .	<b>21</b>
	<i>John Allspaw, with Matt Massie</i>	
	Time Resolution and Retention Concerns	22
	Locality of Metrics Collection and Storage	23
	Layers of Metrics	24
	Providing Context for Anomaly Detection and Alerts	27
	Log Lines Are Metrics, Too	28
	Correlation with Change Management and Incident Time-	
	lines	30
	Making Metrics Available to Your Alerting Mechanisms	31
	Using Metrics to Guide Load-Feedback Mechanisms	32
	A Metrics Collection System, Illustrated: Ganglia	36
	Conclusion	47

<b>4</b>	<b>Continuous Deployment . . . . .</b>	<b>49</b>
	<i>Eric Ries</i>	
	Small Batches Mean Faster Feedback	49
	Small Batches Mean Problems Are Instantly Localized	50
	Small Batches Reduce Risk	50
	Small Batches Reduce Overhead	51
	The Quality Defenders' Lament	52
	Getting Started	56
	Continuous Deployment Is for Mission-Critical Applications	60
	Conclusion	63
<b>5</b>	<b>Infrastructure As Code . . . . .</b>	<b>65</b>
	<i>Adam Jacob</i>	
	Service-Oriented Architecture	67
	Conclusion	79
<b>6</b>	<b>Monitoring . . . . .</b>	<b>81</b>
	<i>Patrick Debois</i>	
	Story: "The Start of a Journey"	81
	Step 1: Understand What You Are Monitoring	85
	Step 2: Understand Normal Behavior	95
	Step 3: Be Prepared and Learn	102
	Conclusion	106
<b>7</b>	<b>How Complex Systems Fail . . . . .</b>	<b>107</b>
	<i>John Allspaw and Richard Cook</i>	
	How Complex Systems Fail	108
	Further Reading	116
<b>8</b>	<b>Community Management and Web Operations . . . . .</b>	<b>117</b>
	<i>Heather Champ and John Allspaw</i>	
<b>9</b>	<b>Dealing with Unexpected Traffic Spikes. . . . .</b>	<b>127</b>
	<i>Brian Moon</i>	
	How It All Started	127
	Alarms Abound	128
	Putting Out the Fire	129
	Surviving the Weekend	130
	Preparing for the Future	131

	CDN to the Rescue	131
	Proxy Servers	132
	Corralling the Stampede	132
	Streamlining the Codebase	133
	How Do We Know It Works?	134
	The Real Test	135
	Lessons Learned	136
	Improvements Since Then	136
<b>10</b>	<b>Dev and Ops Collaboration and Cooperation . . . . .</b>	<b>139</b>
	<i>Paul Hammond</i>	
	Deployment	140
	Shared, Open Infrastructure	144
	Trust	146
	On-Call Developers	148
	Avoiding Blame	153
	Conclusion	155
<b>11</b>	<b>How Your Visitors Feel: User-Facing Metrics . . . . .</b>	<b>157</b>
	<i>Alistair Croll and Sean Power</i>	
	Why Collect User-Facing Metrics?	158
	What Makes a Site Slow?	162
	Measuring Delay	165
	Building an SLA	171
	Visitor Outcomes: Analytics	173
	Other Metrics Marketing Cares About	178
	How User Experience Affects Web Ops	179
	The Future of Web Monitoring	180
	Conclusion	185
<b>12</b>	<b>Relational Database Strategy and Tactics for the Web . .</b>	<b>187</b>
	<i>Baron Schwartz</i>	
	Requirements for Web Databases	188
	How Typical Web Databases Grow	193
	The Yearning for a Cluster	200
	Database Strategy	205
	Database Tactics	212
	Conclusion	218

13	<b>How to Make Failure Beautiful: The Art and Science of Postmortems. . . . .</b>	<b>219</b>
	<i>Jake Loomis</i>	
	The Worst Postmortem	220
	What Is a Postmortem?	221
	When to Conduct a Postmortem	222
	Who to Invite to a Postmortem	223
	Running a Postmortem	223
	Postmortem Follow-Up	224
	Conclusion	226
14	<b>Storage . . . . .</b>	<b>227</b>
	<i>Anoop Nagwani</i>	
	Data Asset Inventory	227
	Data Protection	231
	Capacity Planning	240
	Storage Sizing	242
	Operations	244
	Conclusion	245
15	<b>Nonrelational Databases. . . . .</b>	<b>247</b>
	<i>Eric Florenzano</i>	
	NoSQL Database Overview	248
	Some Systems in Detail	252
	Conclusion	261
16	<b>Agile Infrastructure . . . . .</b>	<b>263</b>
	<i>Andrew Clay Shafer</i>	
	Agile Infrastructure	265
	So, What's the Problem?	269
	Communities of Interest and Practice	279
	Trading Zones and Apologies	279
	Conclusion	282

17	<b>Things That Go Bump in the Night (and How to Sleep Through Them)</b> . . . . .	<b>285</b>
	<i>Mike Christian</i>	
	Definitions	287
	How Many 9s?	288
	Impact Duration Versus Incident Duration	289
	Datacenter Footprint	290
	Gradual Failures	291
	Trust Nobody	292
	Failover Testing	292
	Monitoring and History of Patterns	293
	Getting a Good Night's Sleep	294
	<b>Contributors</b> . . . . .	<b>297</b>
	<b>Index.</b> . . . .	<b>303</b>

# Web Operations: The Career

*Theo Schlossnagle*

**THE INTERNET IS AN INTERESTING MEDIUM IN WHICH TO WORK.** Almost all forms of business are now being conducted on the Internet, and people continue to capitalize on the fact that a global audience is on the other side of the virtual drive-thru window.

The Internet changes so quickly that we rarely have time to cogitate what we're doing and why we're doing it. When it comes to operating the fabric of an online architecture, things move so fast and change so significantly from quarter to quarter that we struggle to stay in the game, let alone ahead of it. This high-stress, overstimulating environment leads to treating the efforts therein as a job without the concept of a career.

What's the difference, you ask? A career is an occupation taken on for a significant portion of one's life, with opportunities for progress. A job is a paid position of regular employment. In other words, a job is just a job.

Although the Internet has been around for more than a single generation at this point, the Web in its current form is still painfully young and is only now breaking past a single generational marker. So, how can you fill a significant portion of your life with a trade that has existed for only a fraction of the time that one typically works in a lifetime? At this point, to have finished a successful career in web operations, you must have been pursuing this art for longer than it has existed. In the end, it is the pursuit that matters. But make no mistake: pursuing a career in web operations makes you a frontiersman.



# Why Does Web Operations Have It Tough?

Web operations has no defined career path; there is no widely accepted standard for progress. Titles vary, responsibilities vary, and title escalation happens on vastly different schedules from organization to organization.

Although the term *web operations* isn't awful, I really don't like it. The captains, superstars, or heroes in these roles are multidisciplinary experts; they have a deep understanding of networks, routing, switching, firewalls, load balancing, high availability, disaster recovery, Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) services, NOC management, hardware specifications, several different flavors of Unix, several web server technologies, caching technologies, database technologies, storage infrastructure, cryptography, algorithms, trending, and capacity planning. The issue is: how can we expect to find good candidates who are fluent in all of those technologies? In the traditional enterprise, you have architects who are broad and shallow paired with a multidisciplinary team of experts who are focused and deep. However, the expectation remains that your "web operations" engineer be both broad and deep: fix your gigabit switch, optimize your database, and guide the overall infrastructure design to meet scalability requirements.

Web operations is broad; I would argue almost unacceptably broad. A very skilled engineer must know every commonly deployed technology at a considerable depth. The engineer is responsible for operating a given architecture within the described parameters (usually articulated in a service-level agreement, or SLA). The problem is that architecture is, by its very definition, everything. Everything from datacenter space, power, and cooling up through the application stack and all the way down to the HTML rendering and JavaScript executing in the browser on the other side of the planet. Big job? Yes. Mind-bogglingly so.

Although I emphatically hope the situation changes, as it stands now there is no education that prepares an individual for today's world of operating web infrastructures—neither academic nor vocational. Instead, identifying computer science programs or other academic programs that instill strong analytical skills provides a good starting point, but to be a real candidate in the field of web operations you need three things:

## A Strong Background in Computing

Because of the broad required understanding of architectural components, it helps tremendously to understand the ins and outs of the computing systems on which all this stuff runs. Processor architectures, memory systems, storage systems, network switching and routing, why Layer 2 protocols work the way they do, HTTP, database concepts...the list could go on for pages. Having the basics down pat is essential in understanding why and how to architect solutions as well as identify brokenness. It is, after all, the foundation on which we build our intelligence. Moreover, an engineering mindset and a basic understanding of the laws of physics can be a great asset.