Extracted from:

Scalable Cloud Ops with Fugue

Declare, Deploy, and Automate the Cloud

This PDF file contains pages extracted from *Scalable Cloud Ops with Fugue*, published by the Pragmatic Bookshelf. For more information or to purchase a paperback or PDF copy, please visit http://www.pragprog.com.

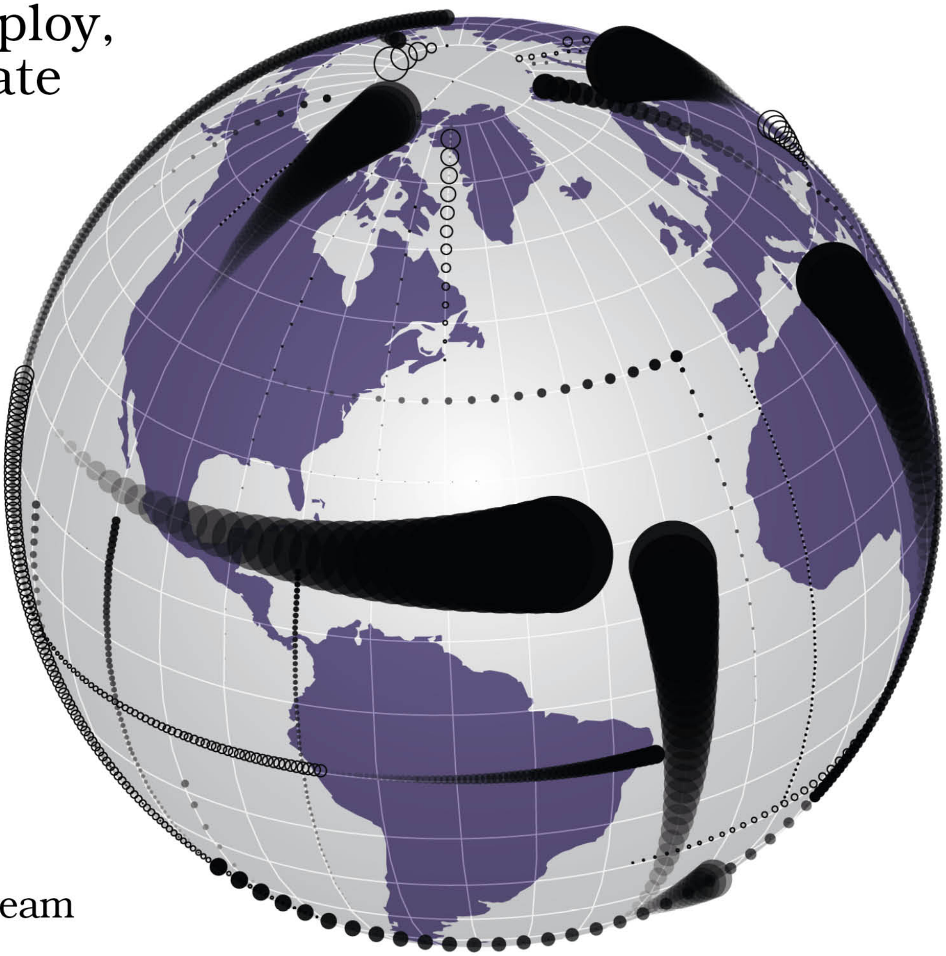
Note: This extract contains some colored text (particularly in code listing). This is available only in online versions of the books. The printed versions are black and white. Pagination might vary between the online and printed versions; the content is otherwise identical.

The Pragmatic Bookshelf
Raleigh, North Carolina

The Pragmatic Programmers

Scalable Cloud Ops with Fugue

Declare, Deploy, and Automate the Cloud



Josha Stella
and the Fugue team

Scalable Cloud Ops with Fugue

Declare, Deploy, and Automate the Cloud

Josha Stella and the Fugue team

The Pragmatic Bookshelf
Raleigh, North Carolina

Pragmatic Bookshelf

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and The Pragmatic Programmers, LLC was aware of a trademark claim, the designations have been printed in initial capital letters or in all capitals. The Pragmatic Starter Kit, The Pragmatic Programmer, Pragmatic Programming, Pragmatic Bookshelf, PragProg and the linking *g* device are trademarks of The Pragmatic Programmers, LLC.

Every precaution was taken in the preparation of this book. However, the publisher assumes no responsibility for errors or omissions, or for damages that may result from the use of information (including program listings) contained herein.

Our Pragmatic books, screencasts, and audio books can help you and your team create better software and have more fun. Visit us at *https://pragprog.com*.

The team that produced this book includes:

Publisher: Andy Hunt
VP of Operations: Janet Furlow
Executive Editor: Susannah Davidson Pfalzer
Development Editor: Katharine Dvorak
Indexing: Potomac Indexing, LLC
Copy Editor: Liz Welch
Layout: Gilson Graphics

For sales, volume licensing, and support, please contact *support@pragprog.com*.

For international rights, please contact *rights@pragprog.com*.

Printed in the United States of America.
ISBN-13: 978-1-68050-234-3
Encoded using the finest acid-free high-entropy binary digits.
Book version: P1.0—July 2017

Prelude

Whether you're just getting started with cloud computing or you're running complex fleets of thousands of instances, this book will help you take the leap from data-center thinking to cloud thinking. Fugue is a system built for operating the cloud, not a system transplanted to the cloud or hacked to work with the cloud. With an infrastructure-as-code approach and an isolated enforcement engine, Fugue simplifies tool, service, and component proliferation in the cloud—an ecosystem that changes frequently and sometimes unpredictably. Fugue solves problems implicit in maintaining infrastructure, in knowing its state, in debugging production systems, and in sweating network configuration details.

The Promise of the Cloud Is Unrealized

The cloud has changed the way we do computing. It liberated us from the burdens of purchasing and maintaining expensive servers and networking hardware. We now consume and scale computing resources as needed and without advanced provisioning. The cloud helps tear down walls between development and operations that slow us down.

But the promise of the cloud is unrealized. Why? Because we still maintain infrastructure in the cloud with the same tools and methods used in the data center. We patch operating systems. We update application code on long-lived instances. We manually configure network changes and those for other services. All this cruft piles up over time, and that's a problem.

Managing and tracking all this change in the cloud is a challenge as instances proliferate and systems become dynamic and distributed. We often launch—and pay for—infrastructure to identify errors. Our deployments fail because systems deviate from what we expect. Application performance suffers due to uneven infrastructure quality. In short, we're wasting time and money.

The Cloud Is the Computer; Fugue Is the Operating System

Cloud computing doesn't have to be this way. The cloud isn't just about shedding the physical data center. It's about shedding the data center mindset. Cloud infrastructure can be controlled via API calls. This means we can view it as a giant general-purpose computer—and program it. This is where Fugue comes in.

Fugue automates the creation, operation, enforcement, and termination of infrastructure in the cloud. It makes immutable patterns practical, improving resilience and security. When an update is needed, Fugue either destroys and replaces a resource (as with a VPC CIDR block) or modifies an existing resource (as with VPC tags), depending on resource properties. The former is an immutable pattern, the latter a mutable one. Fugue's automated enforcement mechanism, which can utilize either pattern, eliminates the configuration drift and unwanted changes that plague production systems.

Fugue comes with Ludwig, a powerful, domain-specific, statically typed language for programming your cloud. Its syntax is easy to compose and read. With its compiler, you identify errors in systems before you spend money building infrastructure. The ability to use libraries and functions saves you time and extends the flexibility of your Fugue compositions, which are accurate, real-time code representations of your infrastructure.

Who Should Read This Book?

This book is for those already using cloud computing or serious about adopting it. You should understand the basic concepts of the cloud and have some experience with the Amazon Web Services (AWS) Management Console before reading this book.

You can use Fugue for all types of cloud-based applications, regardless of size or complexity. Whether you're running production systems on AWS or just sandboxing for now, you'll find this book helpful in your work.

You'll benefit from this book in concrete ways. As you explore the examples, you'll use Fugue to program and launch infrastructure in much less time than it takes with other tools. You'll understand how Fugue enables you to debug in design time, not in the runtime. You'll learn how Fugue automatically and continuously enforces the declared state of your infrastructure.

Fugue removes the need for complex cluster management and inefficient preprovisioning of infrastructure. You can clone and operate consistent environments. You can centralize change control with continuous auditing.

Fugue addresses configuration drift, unwanted changes, infrastructure quality issues, memory leaks due to long-lived components, and errors common in human-performed maintenance. Fugue simplifies DevOps integrations and cloud infrastructure automation for digital enterprises large and small.

How to Read This Book

We suggest starting at the beginning and completing Part I. This will give you an understanding of how Fugue works and a chance to run it in a common application scenario. Once you've completed Part I, feel free to jump to those chapters that are more relevant to your use cases. The book's divided as follows:

Part I—Fugue Computing in Action is all about making the leap from datacenter thinking to cloud thinking. You get an overview of Fugue; take a first step with a simple web app, Fugue Ink; then program, deploy, and maintain a social network application's cloud infrastructure. The latter is introduced via *Refuge*, an imaginary company with namesake software that faces very common and real-world development and production challenges. You'll work along in scenarios of growing complexity, adding services and refactoring for new features. You'll complete the application while continuing to explore Fugue's architectural concepts and language types.

Part II—Tackling More Real-World Complexity covers Fugue's language, Ludwig, in depth and the advantages its abstractions provide for speeding deployment and reducing failure in the cloud. You'll get hands-on experience with an expanded array of Ludwig types that correspond to a wide range of services and use cases. You'll learn how to use functions to better manage infrastructure and how to write custom libraries. Practicing with Ludwig gives you the freedom to do things that are unforeseen, inventive, customized, and both uniquely and practically aligned with your use cases. You'll also learn a great deal about multicomponent coordination and configuration synchronization using our variable service tool, Vars.

Part III—A Cloud-Minded Enterprise will help you stay efficient using Fugue as your team and applications grow globally and your life-cycle management challenges intensify. You'll learn about Fugue's integrations with some well-known tools and collaboration features useful in enterprise workflows. Strategies to avoid accumulating unnecessary technical debt are key. Finally, you'll survey pivotal developments related to operating systems and how those inform Fugue and the cloud's present and future management. The thinking that's shaped where we are now and where cloud-first architecting might lead points us in interesting directions.

This book covers specific AWS services, including EC2 (compute), S3 (storage), RDS (MySQL), DynamoDB (NoSQL), security groups (firewalls), VPC (networking), ASG (Auto Scaling groups), and many others, as well as Fugue's replicated key/value store, Vars. In time, Fugue will become available on additional cloud platforms, like Microsoft Azure. The concepts and examples in this book will carry forward accordingly.

Online Resources

On the Fugue website,[1] you can find the following:

- Comprehensive documentation of Fugue's command-line interface (CLI), Conductor features, Ludwig language tutorials with Fugue compositions, libraries, integration resources, and a wide variety of applied infrastructure examples.
- Product information, with webinars and FAQs.
- A link that leads you to the Fugue Download Portal, which also includes release information.
- A link to this book's companion website at The Pragmatic Bookshelf,[2] where you can download the sample code files to work along with the examples in the book. You can also access the book's online forum, where you will be able to interact with other readers as well as the Fugue team to get the most out of the book. And, you will find a link to report any errata you discover.

Why "Fugue"?

Fugue (/fyoog/) is a rare word, but a universal form. It originally comes to us from music. A fugue spins out a musical theme. Imagine an unwinding that starts small and expands beyond expectation. First comes a short melody. Another part or player takes up that melody. Then, another takes it up. And another. The parts interweave and the fugue evolves. An elegant, layered, mathematical music plays. It builds complexity from simplicity. It has a logic and an art. A fugue begins with that single theme and develops into a powerful, multipart entity. You can listen closely for its individual components or consider the prodigious harmonies of the whole. Trying to do both can be frustrating or fun, depending on which fugue you're listening to, how full the moon is, and how many times your build broke today.

[1] https://fugue.co

[2] https://pragprog.com/book/fugue

Johann Sebastian Bach immortalized fugues. Cognitive scientist Douglas Hofstadter was a fan of Bach's, and he chose the title of his seminal work deliberately: *Gödel, Escher, Bach: An Eternal Golden Braid [Hof99]*. Within that epic of philosophy, an idea-rich dialogue called "The Ant Fugue" addresses "components" and "the whole" as few other works do. Fugue computing takes its inspiration from this work and its references to Bach's music.

The patterns you'll use in Fugue computing and the system that enables those patterns epitomize the fugue metaphor—its expression of one into many, its mastery of component and whole. As you run each service in your application, you declare simple phrases of computing function and infrastructure with Fugue. The software automates their deployment, replay, and interrelationships. It does so while internally strategizing about resources, enforcing configurations, and, optionally, regenerating cloud compute instances. That is a new approach. It enforces system fidelity. It's grounded in the tenets of immutable infrastructure that we'll mine deeply as we go.

All of this may sound a little abstract and unusual, but by the end of this book, you will have a solid understanding of how Fugue works, when to use it, and why its benefits will change your day-to-day for the better.