

Survive Your Success

Programming

Amazon EC2



O'REILLY®

*Jurg van Vliet
& Flavia Paganelli*

Programming Amazon EC2

If you plan to use Amazon Web Services to run applications in the cloud, the end-to-end approach in this book will save you needless trial and error. You'll find practical guidelines for designing and building applications with Amazon Elastic Compute Cloud (EC2) and a host of supporting AWS tools, with a focus on critical issues such as load balancing, monitoring, and automation.

How do you move an existing application to AWS, or design your application so that it scales effectively? How much storage will you require? *Programming Amazon EC2* not only helps you get started, it will also keep you going once you're successfully positioned in the cloud. This book is a must-read for application architects, developers, and administrators.

- Determine your application's lifecycle and identify the AWS tools you need
- Learn how to build and run your application as part of the development process
- Migrate simple web applications to the cloud with EC2, Amazon Simple Storage Service, and CloudFront content delivery
- Meet traffic demand with EC2's Auto Scaling and Elastic Load Balancing
- Decouple your application using Simple Queue Service, Simple Notification Service, and other tools
- Use the right tools to minimize downtime, improve uptime, and manage your decoupled system

“Jurg and Flavia have done a great job in this book building a practical guide on how to build real systems using AWS.”

—Werner Vogels
VP & CTO at Amazon.com

Jurg van Vliet co-founded 9Apps, an AWS solution provider and silver partner of Eucalyptus. He worked for various web startups and media companies before starting the e-learning firm 2Yellows.

Flavia Paganelli, co-founder of 9Apps, has been developing software over the past 14 years for several industries while working at companies such as TomTom and Layar. She is the creator of the Decaf EC2 smartphone app.

Previous programming experience is recommended.

O'REILLY[®]
oreilly.com

US \$34.99

CAN \$40.99

ISBN: 978-1-449-39368-7



Safari[®]
Books Online

Free online edition

for 45 days with purchase of this book. Details on last page.

Programming Amazon EC2

Programming Amazon EC2

Jurg van Vliet and Flavia Paganelli

O'REILLY®
Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo

Programming Amazon EC2

by Jurg van Vliet and Flavia Paganelli

Copyright © 2011 I-MO BV. All rights reserved.

Printed in the United States of America.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (<http://my.safaribooksonline.com>). For more information, contact our corporate/institutional sales department: (800) 998-9938 or corporate@oreilly.com.

Editors: Mike Loukides and Julie Steele

Production Editor: Adam Zaremba

Copyeditor: Amy Thomson

Proofreader: Emily Quill

Indexer: John Bickelhaupt

Cover Designer: Karen Montgomery

Interior Designer: David Futato

Illustrator: Robert Romano

Printing History:

February 2011: First Edition.

Nutshell Handbook, the Nutshell Handbook logo, and the O'Reilly logo are registered trademarks of O'Reilly Media, Inc. *Programming Amazon EC2*, the image of a bushmaster snake, and related trade dress are trademarks of O'Reilly Media, Inc.

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 O'Reilly Media, Inc. was aware of a trademark claim, the designations have been printed in caps or initial caps.

While every precaution has been taken in the preparation of this book, the publisher and authors assume no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein.

ISBN: 978-1-449-39368-7

[LSI]

1297365147

Table of Contents

Foreword	ix
Preface	xiii
1. Introducing AWS	1
From 0 to AWS	1
Biggest Problem First	2
Infinite Storage	3
Computing Per Hour	4
Very Scalable Data Store	5
Optimizing Even More	6
Going Global	7
Growing into Your Application	7
Start with Realistic Expectations	7
Simply Small	8
Growing Up	9
Moving Out	10
“You Build It, You Run It”	11
Individuals and Interactions: One Team	11
Working Software: Shared Responsibility	12
Customer Collaboration: Evolve Your Infrastructure	13
Responding to Change: Saying Yes with a Smile	13
In Short	14
2. Starting with EC2, RDS, and S3/CloudFront	15
Setting Up Your Environment	16
Your AWS Account	16
Command-Line Tools	17
AWS Management Console	19
Other Tools	20
Choosing Your Geographic Location, Regions, and Availability Zones	21

Choosing an Architecture	21
Creating the Rails Server on EC2	22
Creating a Key Pair	23
Finding a Suitable AMI	23
Setting Up the Web/Application Server	24
RDS Database	35
Creating an RDS Instance (Launching the DB Instance Wizard)	36
Is This All?	39
S3/CloudFront	41
Setting Up S3 and CloudFront	41
Static Content to S3/CloudFront	43
Making Backups of Volumes	45
Installing the Tools	46
Running the Script	46
In Short	49
3. Growing with S3, ELB, Auto Scaling, and RDS	51
Preparing to Scale	52
Setting Up the Tools	54
S3 for File Uploads	54
User Uploads for Kulitzer (Rails)	54
Elastic Load Balancing	55
Creating an ELB	56
Difficulties with ELB	59
Auto Scaling	60
Setting Up Auto Scaling	60
Auto Scaling in Production	64
Scaling a Relational Database	66
Scaling Up (or Down)	66
Scaling Out	68
Tips and Tricks	69
Elastic Beanstalk	70
In Short	72
4. Decoupling with SQS, SimpleDB, and SNS	73
SQS	73
Example 1: Offloading Image Processing for Kulitzer (Ruby)	74
Example 2: Priority PDF Processing for Marvia (PHP)	77
Example 3: Monitoring Queues in Decaf (Java)	81
SimpleDB	85
Use Cases for SimpleDB	87
Example 1: Storing Users for Kulitzer (Ruby)	88
Example 2: Sharing Marvia Accounts and Templates (PHP)	91

Example 3: SimpleDB in Decaf (Java)	95
SNS	99
Example 1: Implementing Contest Rules for Kulitzer (Ruby)	100
Example 2: PDF Processing Status (Monitoring) for Marvia (PHP)	105
Example 3: SNS in Decaf (Java)	108
In Short	111
5. Managing the Inevitable Downtime	113
Measure	114
Up/Down Alerts	114
Monitoring on the Inside	114
Monitoring on the Outside	118
Understand	122
Why Did I Lose My Instance?	122
Spikes Are Interesting	122
Predicting Bottlenecks	124
Improvement Strategies	124
Benchmarking and Tuning	124
The Merits of Virtual Hardware	125
In Short	126
6. Improving Your Uptime	129
Measure	129
EC2	130
ELB	131
RDS	132
Using Dimensions from the Command Line	133
Alerts	134
Understand	136
Setting Expectations	136
Viewing Components	137
Improvement Strategies	138
Planning Nonautoscaling Components	138
Tuning Auto Scaling	138
In Short	138
7. Managing Your Decoupled System	141
Measure	141
S3	142
SQS	142
SimpleDB	149
SNS	152
Understand	153

Imbalances	154
Bursts	154
Improvement Strategies	154
Queues Neutralize Bursts	155
Notifications Accelerate	155
In Short	156
8. And Now...	157
Other Approaches	157
Private/Hybrid Clouds	158
Thank You	158
Index	159

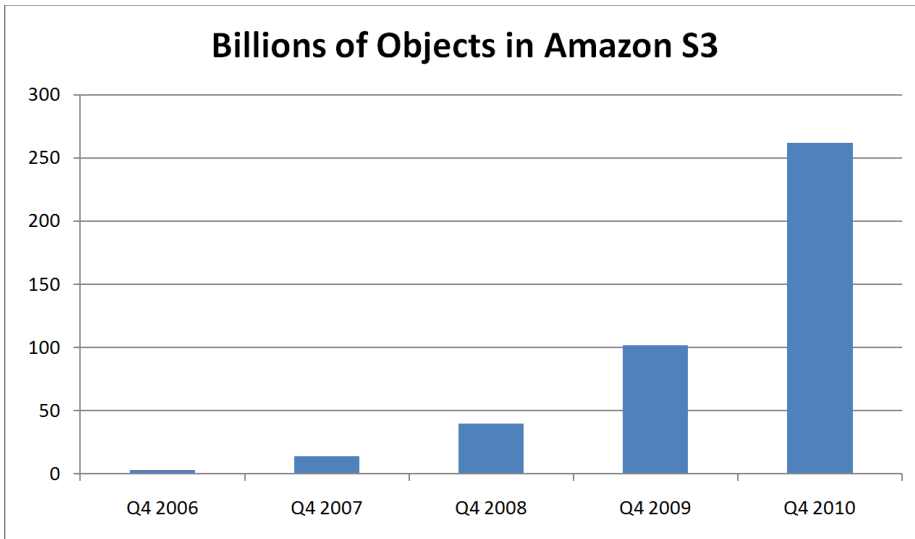
Foreword

March 14, 2006, was an important day, even though it is unlikely that it will ever become more than a footnote in some history books. On that day, Amazon Web Services launched the first of its utility computing services: the Amazon Simple Storage Service (Amazon S3). In my eyes that was the day that changed the way IT was done; it gave everyone access to an ultra-reliable and highly scalable storage service without having to invest tens of thousands of dollars for an exclusive enterprise storage solution. And even better, the service sat directly on the Internet, and objects were directly HTTP addressable.

The motivation behind the launch of the service was simple: the AWS team had asked itself what innovation could happen if it could give everyone access to the same scalable and reliable technologies that were available to Amazon engineers. A student in her dorm room could have an idea that could become the next Amazon or the next Google, and the only thing that would hold her back was access to the resources needed to fulfill that potential. AWS aimed at removing these barriers and constraints so people could unleash their innovation and focus on building great new products instead of having to invest in infrastructure both intellectually and financially.

Today, Amazon S3 has grown to store more than 260 billion objects and routinely runs more than 200,000 storage operations per second. The service has become a fundamental building block for many applications, from enterprise ERP log files to blog storage, streaming videos, software distribution, medical records, and astronomy data.

By routinely running over 200,000 storage operations per second, Amazon S3 is a marvel of technology under the covers. It is designed to support a wide range of usage scenarios and is optimized in very innovative ways to make sure every customer gets great service, regardless of whether he is streaming videos or just housing some home photos. One of my colleagues had a great analogy about how the Amazon S3 software had to evolve: it was like starting with a single-engine Cessna that had to be rebuilt into a Boeing 747 while continuing to fly and continuously refueling, and with passengers that changed planes without noticing it. The Amazon S3 team has done a great job of making the service something millions and millions of people rely on every day.



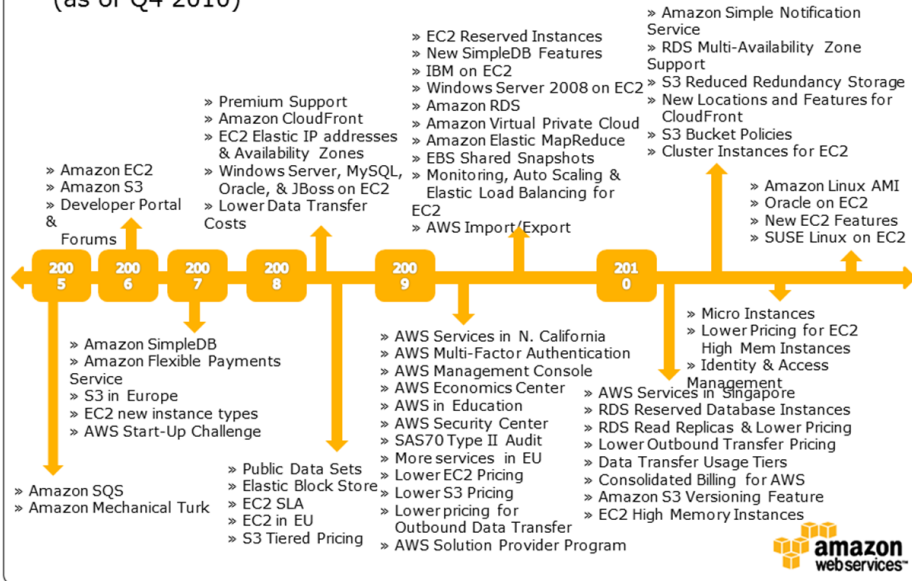
Following Amazon S3, we launched Amazon Simple Queue Service (Amazon SQS), and then Amazon Elastic Compute Cloud (Amazon EC2) just a few months later. These services demonstrated the power of what we have come to call Cloud Computing: access to highly reliable and scalable infrastructure with a utility payment model that drives innovation and dramatically shortens time to market for new products. Many CIOs have told me that while their first motivation to start using AWS was driven by the attractive financial model, the main reason for staying with AWS is that it has made their IT departments agile and allowed them to become enablers of innovation within their organization.

The AWS platform of technology infrastructure services and features has grown rapidly since that day in March 2006, and we continue to keep that same quick pace of innovation and relentless customer focus today.

Although AWS, as well as its ecosystem, has launched many tools that make using the services really simple, at its core it is still a fully programmable service with incredible power, served through an API. Jurg and Flavia have done a great job in this book of building a practical guide for how to build real systems using AWS. Their writing is based on real experiences using each and every one of the AWS services, and their advice is rooted in building foundations upon which applications on the AWS platform can scale and remain reliable. I first came in contact with them when they were building Decaf, an Android application used to control your AWS resources from your mobile device. Since then, I have seen them help countless customers move onto the AWS platform, and also help existing customers scale better and become more reliable while taking advantage of the AWS elasticity to drive costs down. Their strong customer focus makes them great AWS partners.

AWS Pace of Innovation

(as of Q4 2010)



The list of services and features from these past years may seem overwhelming, but our customers continue to ask for more ways to help us remove nonessential infrastructure tasks from their plate so that they can focus on what really matters to them: delivering better products and services to their customers.

AWS will continue to innovate on behalf of our customers, and there are still very exciting things to come.

—Werner Vogels
VP & CTO at Amazon.com

Preface

Thank you for picking up a copy of this book. Amazon Web Services (AWS) has amazed everyone: Amazon has made lots of friends, and all its “enemies” are too busy admiring AWS to do much fighting back. At the moment, there is no comparable public Infrastructure as a Service (IaaS); AWS offers the services at a scale that has not been seen before. We wrote this book so you can get the most out of AWS’ services. If you come from conventional hardware infrastructures, once you are on AWS, you won’t want to go back.

AWS is not easy; it combines skills of several different (established) crafts. It is different from traditional systems administration, and it’s not just developing a piece of software. If you have practiced one or both of these skills, all you need is to be inquisitive and open to learning.

Our background is in software engineering. We are computer scientists with extensive software engineering experience in all sorts of different fields and organizations. But the cloud in general and AWS in particular caught our interest some years ago. We got serious about this by building Decaf, an Android smartphone application that manages Amazon EC2 (Elastic Compute Cloud) accounts. We were finalists in the Android Developer Challenge in 2009. We will use Decaf to illustrate various AWS services and techniques throughout this book.

Around the same time, in early 2010, we decided we wanted to build applications on AWS. We founded 9Apps and set out to find a select group of partners who shared our development interests. Our expertise is AWS, and our responsibility is to keep it running at all times. We design, build, and operate these infrastructures.

Much of our experience comes from working with these teams and building these applications, and we will use several of them as examples throughout the book. Here is a short introduction to the companies whose applications we will use:

Directness

Directness helps customers connect brands to businesses. With a set of tools for making surveys and collecting, interpreting, and presenting consumers’ feedback, this application is very successful in its approach and works with a number of international firms. The problem is scaling the collection of customer responses,

transforming it into usable information, and presenting it to the client. Directness can only grow if we solve this problem.

Kulitzer

Kulitzer is a web application that allows users to organize creative contests. Users can invite participants to enter the contest, an audience to watch, and a jury to pick a winner. Technically, you can consider Kulitzer a classical consumer web app.

Layar

Layar is an augmented reality (AR) smartphone browser that is amazing everyone. This application enriches the user's view of the world by overlapping different objects or information in the camera view, relevant to the location. For example, users can see what people have been tweeting near them, the houses that are for sale in the neighborhood, or tourist attractions near where they are walking.

The Layar application continues to win prize after prize, and is featured in many technical and mainstream publications. Layar started using Google App Engine for its servers, but for several reasons has since moved to AWS.

Marvia

Ever needed to create some “print ready” PDFs? It's not an easy task. You probably needed desktop publishing professionals and the help of a marketing agency, all for a significant price tag. Marvia is an application that can dramatically reduce the effort and cost involved in PDF creation. It allows you to create reusable templates with a drag-and-drop web application. Or you can integrate your own system with Marvia's API to automate the generation of leaflets and other material.

Publitas

Publitas does the opposite of what Marvia does, in a way. It lets you transform your traditional publication material to an online experience. The tool, called ePublisher, is very feature-rich and is attracting a lot of attention. You can input your material in PDF format to the application and it will generate online content. You can then enrich the content with extra functionality, such as supporting sharing in social networks and adding music, video, search, and print. The challenge with the Publitas software is that its existing customers are established and well-known businesses that are sometimes already so powerful that exposure ratings resemble those of a mass medium like television.

Audience

Of course, we welcome all readers of this book, and we hope it inspires you to get into AWS and utilize it in the best possible way to be successful. But we set out to write this book with a particular purpose: to be an AWS guide for building and growing applications from small to “Internet scale.” It will be useful if you want to host your blog or small web application, but it will also help you grow like Zynga did with Farmville. (Some say Zynga is the fastest growing company in the world.)

This book does not focus on detail; for example, we are not going to tell you exactly which parameters each command receives, and we are not going to list all the available commands. But we will show you the approach and implementation. We rely on examples to illustrate the concepts and to provide a starting point for your own projects. We try to give you a sense of all AWS functionality, which would be nearly impossible if we were to show the details of every feature.

To get the most out of this book, you should be comfortable with the command line, and having experience writing software will be useful for some of the chapters. And it certainly wouldn't hurt if you know what Ubuntu is (or CentOS or Windows 2003, for that matter) and how to install software. But most of all, you should simply be curious about what you can do with AWS. There's often more than one way of doing things, and since AWS is so new, many of those ways have not yet been fully explored.

If you are a seasoned software/systems engineer or administrator, there are many things in this book that will challenge you. You might think you know it all. Well, you don't!

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.



This icon signifies a tip, suggestion, or general note.



This icon indicates a warning or caution.


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: "*Programming Amazon EC2* by Jurg van Vliet and Flavia Paganelli. Copyright 2011 I-MO BV, 978-1-449-39368-7."

If you feel your use of code examples falls outside fair use or the permission given above, feel free to contact us at permissions@oreilly.com.

Safari® Books Online

 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>.

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://www.oreilly.com/catalog/9781449393687>

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

bookquestions@oreilly.com

For more information about our books, courses, conferences, and news, see our website at <http://oreilly.com>.

Find us on Facebook: <http://facebook.com/oreilly>

Follow us on Twitter: <http://twitter.com/oreillymedia>

Watch us on YouTube: <http://www.youtube.com/oreillymedia>

Acknowledgments

There are many people we would like to thank for making this book into what it is now. But first of all, it would never have been possible without our parents, Aurora Gómez, Hans van Vliet, Marry van Vliet, and Ricardo Paganelli.

Right from the start we have been testing our ideas with many friends and colleagues; their early feedback shaped this book significantly. Thanks to Adam Dorell, Arjan van Woensel, Björn van Vliet, Dirk Groten, Eduardo Röhr, Eric Hammond, Federico Mikaelian, Fleur van Vliet, Grant Wilson, Indraneel Bommisetty, Joerg Seibel, Khalil Seyedmehdi, Marten Mickos, Matt Miesnieks, Pambo Pascalides, Pedro Moranga Gonçalves, Pim Dernelen, Roy Chandra, Steven van Wel, Werner Vogels, Wouter Broekhof, and Zoran Kovačević.

Of course, you need “strange eyes” going over every detail and meticulously trying out examples to find errors. Our technical reviewers, Anthony Maës, Ben Immanuel, Graziano Obertelli, and Menno van der Sman, did just that.

And finally, there is the wonderful and extremely professional team at O’Reilly. Without Mike, Julie, and all the others there wouldn’t even have been a book. To Amy Thomson, Adam Zaremba, Julie Steele, Mike Loukides, Sumita Mukherji, and the rest we met and worked with, thank you!