# Preface

System administration is a complicated topic that requires practitioners to be familiar with an ever-expanding range of applications and services. In some ways, Amazon Web Services (AWS) is just another tool to add to your toolkit, yet it can also be considered a discipline in and of itself. Successfully building and deploying infrastructure on AWS involves a thorough understanding of the underlying operating system concerns, software architecture, and delivery practices, as well as the myriad components that make up Amazon Web Services.

Mike runs a DevOps consultancy, helping startups and small businesses reap the benefits of tools and processes that were previously available only to organizations with large teams of systems administrators. Many of these businesses do not have a dedicated systems administrator, and the development team is responsible for deploying and maintaining the architecture.

In working with these clients, Mike noticed patterns in how people were working with AWS. Those who came from a pure development background, without any sysadmin experience, would often build an infrastructure that left out many of the things sysadmins would take for granted, such as monitoring and logging. The lack of monitoring and logging would then make it difficult to track down issues, leading to more downtime than necessary.

At the other end of the spectrum were those with a lot of sysadmin experience, but less or no development experience. This group was more likely to treat AWS as nothing more than a virtual machine hosting provider, simply using EC2 to run a fleet of static instances without taking advantage of any high-availability features such as Auto Scaling groups and Elastic Load Balancing. This is akin to buying a Ferrari and then using it only to buy groceries once per week: fun, but not very cost-effective.

Using AWS requires a fundamentally different mindset than when deploying groups of static servers. You do not simply set up some servers and then periodically perform maintenance. Instead, you use the full AWS toolset (automatic instance replacement, scaling up and down in response to demand, etc.) to build a system. In this sense, it is more like programming than traditional system administration.

Federico’s work as the lead Product Manager for Ubuntu Server at Canonical placed him on the front lines from the vendor perspective. Looking at how users and public cloud vendors were integrating Ubuntu into their infrastructure informed the team’s decisions as to what user problems to solve first, and led to the creation of Canonical’s Certified Public Cloud program, which solves integration problems for public cloud vendors and manages the relationship between Ubuntu and the Amazon AWS technical team.

The Ubuntu Server team’s cloud-first focus led to the creation of technologies like Cloud-Init, and produced the early, smooth integration of Ubuntu with public cloud that resulted in its popularity there today. Federico’s aim has been to complement Mike’s knowledge as a power user of public cloud with his behind-the-scenes insight of where things can go wrong for users.

The aim of this book is to help you reach a balance between development and operational focus, and help you make the right choice for your application’s specific hosting requirements. If you are a developer, this book will give you enough system administration knowledge to ensure that you are using AWS effectively, and help you build a robust and resilient application infrastructure. For systems administrators, it will show you how you can keep your favorite tools and processes while working with AWS, and hopefully save you from reinventing some wheels along the way.

AWS is a collection of cloud computing services that can be combined to build scalable and reliable applications and services. It comprises a number of components, each with their own names and configuration options, which are offered under the AWS umbrella. Some of these—such as EC2 and S3—are extremely popular and well-known. Others, such as Kinesis and CloudFormation, are less well-known. Because covering each of these services in detail would result in a multivolume tome of formidable size, this book focuses on the more commonly used services and provides jumping-off points for learning about the others.

If you are familiar with AWS, feel free to hop between chapters to find the information that is most interesting or relevant to your current project. Beginners to AWS should work through the book sequentially, as each chapter builds on information presented in the previous chapters.

[Chapter 1](https://www.safaribooksonline.com/library/view/aws-system-administration/9781449342562/ch01.html#ch_setupawstools) helps you get set up with the tools you will need to interact with AWS and build the example infrastructure.

[Chapter 2](https://www.safaribooksonline.com/library/view/aws-system-administration/9781449342562/ch02.html#chap_first_steps) introduces what is perhaps the most well-known of all AWS services, EC2. This chapter also introduces our favorite AWS service, CloudFormation.

In [Chapter 3](https://www.safaribooksonline.com/library/view/aws-system-administration/9781449342562/ch03.html#chap_aws_access_security), we look at some of the security features offered by AWS.

[Chapter 4](https://www.safaribooksonline.com/library/view/aws-system-administration/9781449342562/ch04.html#aws_config_management) introduces configuration management tools, a common requirement when automating a cloud infrastructure. Using these tools, Chapters [5](https://www.safaribooksonline.com/library/view/aws-system-administration/9781449342562/ch05.html#example_application_stack) and [6](https://www.safaribooksonline.com/library/view/aws-system-administration/9781449342562/ch06.html#aws_sysadmin_autoscaling) demonstrate the process of deploying an example application to AWS.

[Chapter 7](https://www.safaribooksonline.com/library/view/aws-system-administration/9781449342562/ch07.html#aws_deployment) looks at some of the methods of deploying application and infrastructure updates to your environment. [Chapter 8](https://www.safaribooksonline.com/library/view/aws-system-administration/9781449342562/ch08.html#aws_reusable) builds on this and discusses the creation of reusable components to save time.

Log management, a more traditional sysadmin task that has some interesting implications in the cloud, is the topic of [Chapter 9](https://www.safaribooksonline.com/library/view/aws-system-administration/9781449342562/ch09.html#aws_log).

[Chapter 10](https://www.safaribooksonline.com/library/view/aws-system-administration/9781449342562/ch10.html#aws_dns) covers another traditional sysadmin task: DNS with Amazon’s Route 53 service.

Monitoring with Amazon’s CloudWatch service and other monitoring tools is discussed in [Chapter 11](https://www.safaribooksonline.com/library/view/aws-system-administration/9781449342562/ch11.html#aws_monitoring).

Finally, [Chapter 12](https://www.safaribooksonline.com/library/view/aws-system-administration/9781449342562/ch12.html#aws_backup) looks at some of the ways of backing up your data both in and outside the Amazon cloud.

# Audience

This book is written for system administrators and developers. We assume you are comfortable with the basic tools used to administer the operating system and common services such as DNS. If you plan to use Puppet or Chef for automation, you need to learn basic information about those tools elsewhere. You should have a working knowledge of Git or another source code management system. We do not expect you to have prior knowledge of AWS or other virtualization and cloud products.

Where an operating system choice was required, we tested our examples on Ubuntu 14.04 “Trusty”. Ubuntu is the most popular selection of users on Amazon AWS today, so we used Ubuntu Trusty as both client environment and for our EC2 instances; examples will nonetheless run on most Linux distributions with minimal or no change. We also checked our work against Amazon Linux and Ubuntu 12.04 “Precise” whenever possible, which remain popular choices as we write.

# 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, data types, and environment variables.

**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.

###### NOTE

This icon signifies a general note.

###### TIP

This icon signifies a tip or suggestion.

###### CAUTION

This icon indicates a warning or caution.

# Using Code Examples

This book is here to help you get your job done. Major examples can be downloaded from my [GitHub repository](https://github.com/mikery/aws-system-administration-resources). Many other small examples are scattered through the book; I have not bothered to include them in the repository because they are fairly easy to type in.

In general, you may use the code 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.