contents

preface xiii
acknowledgments xv
about this book xviii
about the authors xx
about the cover illustration xxii

PART 1 FIRST STEPS 1

1 ***Going serverless 3***

1.1 What's in a name? 4
1.2 Understanding serverless architectures 5
Service-oriented architecture and microservices 7 ▪ Implementing architecture the conventional way 7 ▪ Implementing architecture the serverless way 9
1.3 Making the call to go serverless 11
1.4 Serverless pros and cons 14
1.5 What's new in this second edition? 16

2 ***First steps to serverless 18***

2.1 Building a video-encoding pipeline 19
A quick note on AWS costs 19 ▪ Using Amazon Web Services (AWS) 20

2.2 Preparing your system 21

Setting up your system 22 ▪ Working with Identity and Access Management (IAM) 22 ▪ Let's make a bucket 25 ▪ Creating an IAM role 26 ▪ Using AWS Elemental MediaConvert 28 Using MediaConvert Role 29

2.3 Starting with the Serverless Framework 29

Setting up the Serverless Framework 29 ▪ Bringing Serverless Framework to The 24-Hour Video 31 ▪ Creating your first Lambda function 33

2.4 Testing in AWS 36

2.5 Looking at logs 37

3 ***Architectures and patterns 40***

3.1 Use cases 40

Backend compute 41 ▪ Internet of Things (IoT) 41 ▪ Data processing and manipulation 42 ▪ Real-time analytics 42 Legacy API proxy 43 ▪ Scheduled services 44 ▪ Bots and skills 44 ▪ Hybrids 44

3.2 Patterns 45

GraphQL 45 ▪ Command pattern 46 ▪ Messaging pattern 47 ▪ Priority queue pattern 49 ▪ Fan-out pattern 50 Compute as glue 51 ▪ Pipes and filters pattern 52

PART 2 USE CASES 55

4 ***Yubl: Architecture highlights, lessons learned 57***

4.1 The original Yubl architecture 58

Scalability problems 59 ▪ Performance problems 59 ▪ Long feature delivery cycles 59 ▪ Why serverless? 60

4.2 The new serverless Yubl architecture 61

Rearchitecting and rewriting 62 ▪ The new search API 62

4.3 Migrating to new microservices gracefully 64

5 ***A Cloud Guru: Architecture highlights, lessons learned 70***

5.1 The original architecture 71

The journey to 43 microservices 75 ▪ What is GraphQL 77 Moving to GraphQL 79 ▪ Service discovery 80 ▪ Security in the BFF world 82

5.2 Remnants of the legacy 82

6 *Yle: Architecture highlights, lessons learned 84*

6.1 Ingesting events at scale with Fargate 85

Cost considerations 85 ▪ Performance considerations 85

6.2 Processing events in real-time 86

Kinesis Data Streams 86 ▪ SQS dead-letter queue (DLQ) 87 The Router Lambda function 88 ▪ Kinesis Data Firehose 88 Kinesis Data Analytics 89 ▪ Putting it altogether 90

6.3 Lessons learned 91

Know your service limits 91 ▪ Build with failure in mind 93 Batching is good for cost and efficiency 94 ▪ Cost estimation is tricky 95

PART 3 PRACTICUM 97

7 *Building a scheduling service for ad hoc tasks 99*

7.1 Defining nonfunctional requirements 101

7.2 Cron job with EventBridge 102

Your scores 104 ▪ Our scores 105 ▪ Tweaking the solution 107 ▪ Final thoughts 109

7.3 DynamoDB TTL 109

Your scores 110 ▪ Our scores 111 ▪ Final thoughts 113

7.4 Step Functions 113

Your scores 115 ▪ Our scores 115 ▪ Tweaking the solution 116 ▪ Final thoughts 119

7.5 SQS 119

Your scores 120 ▪ Our scores 120 ▪ Final thoughts 122

7.6 Combining DynamoDB TTL with SQS 122

Your scores 123 ▪ Our scores 124 ▪ Final thoughts 125

7.7 Choosing the right solution for your application 125

7.8 The applications 125

Your weights 126 ▪ Our weights 126 ▪ Scoring the solutions for each application 128

8 *Architecting serverless parallel computing 132*

8.1 Introduction to MapReduce 133

How to transcode a video 134 ▪ Architecture overview 135

8.2 Architecture deep dive 137

Maintaining state 138 ▪ Step Functions 141

8.3 An alternative architecture 144

9 ***Code Developer University 146***

9.1 Solution overview 147

Requirements listed 147 ▪ Solution architecture 148

9.2 The Code Scoring Service 150

Submissions Queue 152 ▪ Code Scoring Service summary 153

9.3 Student Profile Service 153

Update Student Scores function 155

9.4 Analytics Service 157

Kinesis Firehose 158 ▪ AWS Glue and Amazon Athena 160 QuickSight 163

PART 4 THE FUTURE 165

10 ***Blackbelt Lambda 167***

10.1 Where to optimize? 167

10.2 Before we get started 169

How a Lambda function handles requests 169 ▪ Latency: Cold vs. warm 173 ▪ Load generation on your function and application 173 ▪ Tracking performance and availability 174

10.3 Optimizing latency 176

Minimize deployment artifact size 176 ▪ Allocate sufficient resources to your execution environment 178 ▪ Optimize function logic 179

10.4 Concurrency 180

Correlation between requests, latency, and concurrency 181 Managing concurrency 181

11 ***Emerging practices 183***

11.1 Using multiple AWS accounts 184

Isolate security breaches 184 ▪ Eliminate contention for shared service limits 185 ▪ Better cost monitoring 185 ▪ Better autonomy for your teams 185 ▪ Infrastructure-as-code for AWS Organizations 186

11.2 Using temporary stacks 186

Common AWS account structure 186 ▪ Use temporary stacks for feature branches 187 ▪ Use temporary stacks for e2e tests 188

11.3 Avoid sensitive data in plain text in environment variables 188

Attackers can still get in 189 ▪ Handle sensitive data securely 189

11.4 Use EventBridge in event-driven architectures 190

Content-based filtering 190 ▪ Schema discovery 191 ▪ Archive and replay events 191 ▪ More targets 192 ▪ Topology 192

appendix A Services for your serverless architecture 195
appendix B Setting up your cloud 200
appendix C Deployment frameworks 212

index 225