
Table of Contents

Preface.....	xv
--------------	----

Part I. Going Cloud Native

1. What Is a “Cloud Native” Application?.....	3
The Story So Far	4
What Is Cloud Native?	6
Scalability	6
Loose Coupling	7
Resilience	8
Manageability	10
Observability	11
Why Is Cloud Native a Thing?	12
Summary	13
2. Why Go Rules the Cloud Native World.....	15
The Motivation Behind Go	15
Features for a Cloud Native World	16
Composition and Structural Typing	16
Comprehensibility	18
CSP-Style Concurrency	19
Fast Builds	20
Linguistic Stability	21
Memory Safety	22
Performance	22

Static Linking	23
Static Typing	24
Summary	25

Part II. Cloud Native Go Constructs

3. Go Language Foundations.....	29
Basic Data Types	30
Booleans	30
Simple Numbers	31
Complex Numbers	31
Strings	32
Variables	32
Short Variable Declarations	33
Zero Values	33
The Blank Identifier	35
Constants	35
Container Types: Arrays, Slices, and Maps	36
Arrays	36
Slices	37
Maps	41
Pointers	42
Control Structures	44
Fun with for	44
The if Statement	47
The switch Statement	48
Error Handling	49
Creating an Error	50
Putting the Fun in Functions: Variadics and Closures	50
Functions	50
Variadic Functions	54
Anonymous Functions and Closures	55
Structs, Methods, and Interfaces	57
Structs	57
Methods	58
Interfaces	59
Composition with Type Embedding	61
The Good Stuff: Concurrency	64
Goroutines	64

Channels	64
Select	67
Summary	69
4. Cloud Native Patterns.....	71
The Context Package	72
What Context Can Do for You	73
Creating Context	74
Defining Context Deadlines and Timeouts	74
Defining Request-Scoped Values	75
Using a Context	75
Layout of this Chapter	76
Stability Patterns	77
Circuit Breaker	77
Debounce	80
Retry	84
Throttle	86
Timeout	90
Concurrency Patterns	92
Fan-In	93
Fan-Out	95
Future	97
Sharding	101
Summary	106
5. Building a Cloud Native Service.....	107
Let's Build a Service!	107
What's a Key-Value Store?	108
Requirements	108
What Is Idempotence and Why Does It Matter?	108
The Eventual Goal	110
Generation 0: The Core Functionality	110
Your Super Simple API	111
Generation 1: The Monolith	112
Building an HTTP Server with net/http	112
Building an HTTP Server with gorilla/mux	114
Building a RESTful Service	117
Making Your Data Structure Concurrency-Safe	121
Generation 2: Persisting Resource State	123
What's a Transaction Log?	125
Storing State in a Transaction Log File	126

Storing State in an External Database	137
Generation 3: Implementing Transport Layer Security	145
Transport Layer Security	146
Private Key and Certificate Files	147
Securing Your Web Service with HTTPS	148
Transport Layer Summary	149
Containerizing Your Key-Value Store	150
Docker (Absolute) Basics	151
Building Your Key-Value Store Container	158
Externalizing Container Data	162
Summary	163

Part III. The Cloud Native Attributes

6. It's All About Dependability.....	167
What's the Point of Cloud Native?	168
It's All About Dependability	168
What Is Dependability and Why Is It So Important?	169
Dependability: It's Not Just for Ops Anymore	171
Achieving Dependability	172
Fault Prevention	174
Fault Tolerance	176
Fault Removal	176
Fault Forecasting	178
The Continuing Relevance of the Twelve-Factor App	178
I. Codebase	179
II. Dependencies	179
III. Configuration	180
IV. Backing Services	182
V. Build, Release, Run	183
VI. Processes	184
VII. Data Isolation	184
VIII. Scalability	185
IX. Disposability	186
X. Development/Production Parity	186
XI. Logs	187
XII. Administrative Processes	188
Summary	189

7. Scalability.....	191
What Is Scalability?	192
Different Forms of Scaling	193
The Four Common Bottlenecks	194
State and Statelessness	195
Application State Versus Resource State	196
Advantages of Statelessness	196
Scaling Postponed: Efficiency	197
Efficient Caching Using an LRU Cache	198
Efficient Synchronization	201
Memory Leaks Can...fatal error: runtime: out of memory	206
On Efficiency	209
Service Architectures	209
The Monolith System Architecture	210
The Microservices System Architecture	211
Serverless Architectures	213
Summary	217
8. Loose Coupling.....	219
Tight Coupling	220
Tight Coupling Takes Many Forms	221
Communications Between Services	224
Request-Response Messaging	224
Common Request-Response Implementations	225
Issuing HTTP Requests with net/http	226
Remote Procedure Calls with gRPC	230
Loose Coupling Local Resources with Plug-ins	241
In-Process Plug-ins with the plugin Package	241
HashiCorp's Go Plug-in System over RPC	247
Hexagonal Architecture	255
The Architecture	255
Implementing a Hexagonal Service	256
Summary	264
9. Resilience.....	265
Keeping on Ticking: Why Resilience Matters	266
What Does It Mean for a System to Fail?	267
Building for Resilience	268
Cascading Failures	269
Preventing Overload	270

Play It Again: Retrying Requests	275
Backoff Algorithms	276
Circuit Breaking	280
Timeouts	281
Idempotence	286
Service Redundancy	290
Designing for Redundancy	291
Autoscaling	293
Healthy Health Checks	294
What Does It Mean for an Instance to Be “Healthy”?	295
The Three Types of Health Checks	295
Failing Open	300
Summary	300
10. Manageability.....	303
What Is Manageability and Why Should I Care?	304
Configuring Your Application	306
Configuration Good Practice	307
Configuring with Environment Variables	307
Configuring with Command-Line Arguments	308
Configuring with Files	314
Viper: The Swiss Army Knife of Configuration Packages	329
Feature Management with Feature Flags	334
The Evolution of a Feature Flag	334
Generation 0: The Initial Implementation	335
Generation 1: The Hard-Coded Feature Flag	335
Generation 2: The Configurable Flag	336
Generation 3: Dynamic Feature Flags	337
Summary	341
11. Observability.....	343
What Is Observability?	344
Why Do We Need Observability?	345
How Is Observability Different from “Traditional” Monitoring?	345
The “Three Pillars of Observability”	346
OpenTelemetry	347
The OpenTelemetry Components	348
Tracing	350
Tracing Concepts	350
Tracing with OpenTelemetry	352
Putting It All Together: Tracing	363

Metrics	369
Push Versus Pull Metric Collection	371
Metrics with OpenTelemetry	374
Putting It All Together: Metrics	384
Logging	387
Better Logging Practices	388
Logging with Go's Standard log Package	391
The Zap Logging Package	394
Summary	400
Index.....	403

