Kevin Hoffman
Dan Nemeth

Cloud Native Go

Building Web Applications and Microservices for the Cloud with Go and React

Hong Kong Skyline & Harbor

The cover image, by Lee Yiu Tung, shows a portion of the Hong Kong skyline and harbor. According to The Skyscraper Center, Hong Kong is home to 315 buildings at least 150 meters in height: more than any other city on Earth. Nearly three-fourths of Hong Kong's skyscrapers are residential, helping to explain why more residents live above the 14th floor than in any other city. Hong Kong's tallest building, the International Commerce Centre, is 484 meters high—more than 40 meters taller than the tip of the Empire State Building's spire. At night, during good weather, visitors can experience "A Symphony of Lights," a light and laser show incorporating dozens of buildings on each side of Hong Kong's Victoria Harbor. The Harbor itself—still named after Britain's Queen Victoria nearly 20 years after Hong Kong was restored to China—holds 263 islands, as well as watercraft ranging from cargo freighters to cruise ships, and tourist ferries to traditional Chinese sampans and junks.

Cloud Native Go: Building Web Applications and Microservices for the Cloud with Go and React

Table of Contents

- Cover
- Title Page
- Copyright Page
- About the Authors
- Acknowledgments
- Contents
- 1 The Way of the Cloud
  - The Virtues of the Way of the Cloud
    - Favor Simplicity
    - Test First, Test Everything
    - Release Early, Release Often
    - Automate Everything
    - Build Service Ecosystems
  - Why Use Go?
    - Simplicity
    - Open Source
    - Easy Automation and IDE Freedom
  - Summary
- 2 Getting Started
  - The Right Tools for the Job

Table of Contents

Setting Up Git

Installing Homebrew

Installing the Git Client

Installing Mercurial and Bazaar

Creating a GitHub Account

Setting Up Go

Configuring your Go Workspace

Testing Your Environment

Summary

3 Go Primer

Building Hello Cloud

Using Basic Functions

Working with Structs

Introducing Go Interfaces

Adding Methods to Structs

Exploiting Dynamic Typing in Go with Interfaces

Working with Third-Party Packages

Creating Your Own Packages

Exporting Functions and Data

Creating a Package

Summary

4 Delivering Continuously

Introducing Docker

Why use Docker?

Installing Docker

Running Docker Images

Continuous Integration with Wercker

Table of Contents

Continuous Integration Best Practices

Why use Wercker?

Creating a Wercker Application

Installing the Wercker CLI

Creating a Wercker Configuration File

Running a Wercker Build

Deploying to DockerHub

Reader Exercise: Create a Full Development Pipeline

Advanced Challenge: Integrate Third-Party Library

Summary

5 Building Microservices in Go

Designing Services API First

Designing the Matches API

Creating an API Blueprint

Testing and Publishing Documentation with Apiary

Scaffolding a Microservice

Building Services Test First

Creating a First, Failing Test

Testing the Location Header

Epic MontageTest Iterations

Deploying and Running in the Cloud

Creating a PWS Account

Setting up PCF Dev

Pushing to Cloud Foundry

Summary

6 Using Backing Services

Designing Service Ecosystems

Building Dependent Services Test-First

Table of Contents

Creating the Fulfillment Service

Creating the Catalog Service

Sharing Structured Data Among Services

Client Imports Server Package

Client Duplicates Server Structure

Client And Server Import Shared Package

Using Service Bindings to Externalize URLs and Metadata

Discovering Services

Dynamic Service Discovery

Service Discovery with Netflixs Eureka

Reader Exercise

Bonus Activity

Summary

7 Creating a Data Service

Creating a MongoDB Match Repository

Why MongoDB?

Updating the Repository Pattern

Communicating with MongoDB in Go

Writing a MongoDB Repository Test-First

Integration Testing a Mongo-Backed Service

Integrating with a Transient MongoDB Database

Writing an Integration Test

Running in the Cloud

Configuring a Backing Service

Summary

8 Event Sourcing and CQRS

Reality is Event Sourced

Table of Contents

Idempotent

Isolated

Testable

Replayable and Recoverable

Big Data

Embracing Eventual Consistency

Introducing Command Query Responsibility Segregation

Event Sourcing Use Cases

Weather Monitoring

Connected Cars

Social Media Feed Processing

Code Sample: Managing a Drone Army

Creating a Command Handler Service

Introduction to RabbitMQ

Building the Command Handler Service

Building the Event Processor

Integration Testing the Event Processor

Creating the Query Handler Service

Summary

9 Building a Web Application with Go

Serving Static Files and Assets

Supporting JavaScript Clients

Using Server-Side Templates

Processing Forms

Working with Cookies and Session State

Writing Cookies

Reading Cookies

Table of Contents

Build and Deploy with Wercker

Summary

10 Security in the Cloud

Securing a Web Application

Web Application Security Options

Setting up an Auth0 Account

Building an OAuth-Secured Web App

Running the SecureWeb Application

Securing Microservices

Overview of the Client Credentials Pattern

Securing a Microservice with Client Credentials

A Note on SSL

A Word on Privacy and Data Security

Hackers Cant Get What You Dont Have

Reader Exercise

Summary

11 Working with WebSockets

Demystifying WebSockets

How WebSockets Work

WebSockets vs. Server-Sent Events

Designing a WebSocket Server

On WebSockets and Cloud Nativity

Building a WebSocket App with a Messaging Provider

A Note on JavaScript Frameworks

Running the WebSockets Sample

Summary

12 Building Web Views with React

Table of Contents

- JavaScript State of the Union
- Why React?
  - The Virtual DOM
  - Component Composition
  - Reactive Data Flow
  - Narrow Focus
  - Ease* of Use
- Anatomy of a React Application
  - The package.json File
  - Webpack.config.js
  - The .babelrc File
  - Understanding JSX and Webpack
  - React Components
- Building a Simple React Application
  - What We Didnt Like
- Testing React Applications
- Further Reading
  - React Websites
  - React Books
  - Other Materials
- Summary

13 Creating UIs that Scale with Flux

- Introducing Flux
  - Dispatcher
  - Store
  - View
  - Action
  - Source

Table of Contents

Complexity in Flux

Building a Flux Application

Summary

14 Creating a Full ApplicationWorld of FluxCraft

Introducing World of FluxCraft

Architectural Overview

Independent Scaling, Versioning, and Deployment

Databases are not the Integration Tier

One-Way Immutable Data Flow

The Flux GUI

The Go UI Host Service

Sample Player Move Sequence

Processing Commands

Processing Events

Maintaining Reality

Map Management

Automating Acceptance Tests

Summary

15 Conclusion

What we Learned

Go Isnt Just a Niche Language

How Micro Should a Microservice Be?

Continuous Delivery and Deployment

Test Everything

Release Early, Release Often

Event Sourcing, CQRS, and More Acronyms

Next Steps

Table of Contents

A: Troubleshooting Cloud Applications
Index