

VOLUME 2



# THE PRACTICE OF CLOUD SYSTEM ADMINISTRATION

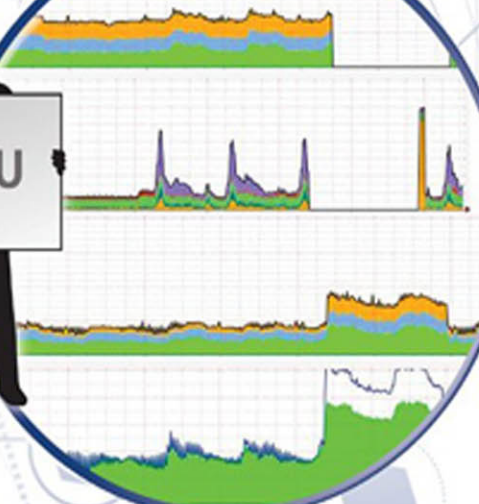
DESIGNING AND OPERATING  
LARGE DISTRIBUTED SYSTEMS

```
subject = subject.String()
change {
  notify in func(notifications map[string]bool) {
    for n, v := range notifications {
      n.Notify(state, n)
    }
  }
  switch status {
    case stCritical:
      notify(stCriticalNotification)
    case stWarn:
      notify(stWarnNotification)
  }
}
if changed {
  s.nc <- true
}

func (s *Scheduler) CheckAlert(a *conf.Alert) {
  crits := s.CheckExpr(a.Crit, stCritical, nil)
  s.CheckExpr(a.Warn, stWarn, crits)
}

func (s *Scheduler) CheckExpr(a *conf.Alert, n *expr.Expr) {
  if a == nil {
    return
  }
  results, err := s.Execute(s.cache, nil)
  if err != nil {
    log.Println(err)
    return
  }
  loop:
  for _, r := range results {
    if a.Squelcher(r.Group) {
      continue
    }
    ak := AlertKey(a.Name, r.Group.String())
    for _, v := range ignore {
      if ak == v {
        continue loop
      }
    }
    results[ak]
  }
}
```

YOU



THOMAS A. LIMONCELLI • STRATA R. CHALUP • CHRISTINA J. HOGAN

# The Practice of Cloud System Administration

Volume 2

# **Practice of Cloud System Administration, The: DevOps and SRE Practices for Web Services, Volume 2**

## **Table of Contents**

Cover

Title Page

Copyright Page

Contents

Preface

About the Authors

Introduction

Part I: Design: Building It

1 Designing in a Distributed World

1.1 Visibility at Scale

1.2 The Importance of Simplicity

1.3 Composition

1.4 Distributed State

1.5 The CAP Principle

1.6 Loosely Coupled Systems

1.7 Speed

1.8 Summary

Exercises

2 Designing for Operations

2.1 Operational Requirements

2.2 Implementing Design for Operations

# **Table of Contents**

2.3 Improving the Model

2.4 Summary

Exercises

## **3 Selecting a Service Platform**

3.1 Level of Service Abstraction

3.2 Type of Machine

3.3 Level of Resource Sharing

3.4 Colocation

3.5 Selection Strategies

3.6 Summary

Exercises

## **4 Application Architectures**

4.1 Single-Machine Web Server

4.2 Three-Tier Web Service

4.3 Four-Tier Web Service

4.4 Reverse Proxy Service

4.5 Cloud-Scale Service

4.6 Message Bus Architectures

4.7 Service-Oriented Architecture

4.8 Summary

Exercises

## **5 Design Patterns for Scaling**

5.1 General Strategy

5.2 Scaling Up

5.3 The AKF Scaling Cube

5.4 Caching

5.5 Data Sharding

5.6 Threading

5.7 Queueing

5.8 Content Delivery Networks

# **Table of Contents**

5.9 Summary

Exercises

## **6 Design Patterns for Resiliency**

6.1 Software Resiliency Beats Hardware Reliability

6.2 Everything Malfunctions Eventually

6.3 Resiliency through Spare Capacity

6.4 Failure Domains

6.5 Software Failures

6.6 Physical Failures

6.7 Overload Failures

6.8 Human Error

6.9 Summary

Exercises

## **Part II: Operations: Running It**

### **7 Operations in a Distributed World**

7.1 Distributed Systems Operations

7.2 Service Life Cycle

7.3 Organizing Strategy for Operational Teams

7.4 Virtual Office

7.5 Summary

Exercises

### **8 DevOps Culture**

8.1 What Is DevOps?

8.2 The Three Ways of DevOps

8.3 History of DevOps

8.4 DevOps Values and Principles

8.5 Converting to DevOps

8.6 Agile and Continuous Delivery

8.7 Summary

Exercises

# **Table of Contents**

## **9 Service Delivery: The Build Phase**

- 9.1 Service Delivery Strategies
- 9.2 The Virtuous Cycle of Quality
- 9.3 Build-Phase Steps
- 9.4 Build Console
- 9.5 Continuous Integration
- 9.6 Packages as Handoff Interface
- 9.7 Summary

Exercises

## **10 Service Delivery: The Deployment Phase**

- 10.1 Deployment-Phase Steps
- 10.2 Testing and Approval
- 10.3 Operations Console
- 10.4 Infrastructure Automation Strategies
- 10.5 Continuous Delivery
- 10.6 Infrastructure as Code
- 10.7 Other Platform Services
- 10.8 Summary

Exercises

## **11 Upgrading Live Services**

- 11.1 Taking the Service Down for Upgrading
- 11.2 Rolling Upgrades
- 11.3 Canary
- 11.4 Phased Roll-outs
- 11.5 Proportional Shedding
- 11.6 Blue-Green Deployment
- 11.7 Toggling Features
- 11.8 Live Schema Changes
- 11.9 Live Code Changes
- 11.10 Continuous Deployment

# **Table of Contents**

11.11 Dealing with Failed Code Pushes

11.12 Release Atomicity

11.13 Summary

Exercises

## **12 Automation**

12.1 Approaches to Automation

12.2 Tool Building versus Automation

12.3 Goals of Automation

12.4 Creating Automation

12.5 How to Automate

12.6 Language Tools

12.7 Software Engineering Tools and Techniques

12.8 Multitenant Systems

12.9 Summary

Exercises

## **13 Design Documents**

13.1 Design Documents Overview

13.2 Design Document Anatomy

13.3 Template

13.4 Document Archive

13.5 Review Workflows

13.6 Adopting Design Documents

13.7 Summary

Exercises

## **14 Oncall**

14.1 Designing Oncall

14.2 Being Oncall

14.3 Between Oncall Shifts

14.4 Periodic Review of Alerts

14.5 Being Paged Too Much

# **Table of Contents**

14.6 Summary

Exercises

## **15 Disaster Preparedness**

15.1 Mindset

15.2 Individual Training: Wheel of Misfortune

15.3 Team Training: Fire Drills

15.4 Training for Organizations: Game Day/DiRT

15.5 Incident Command System

15.6 Summary

Exercises

## **16 Monitoring Fundamentals**

16.1 Overview

16.2 Consumers of Monitoring Information

16.3 What to Monitor

16.4 Retention

16.5 Meta-monitoring

16.6 Logs

16.7 Summary

Exercises

## **17 Monitoring Architecture and Practice**

17.1 Sensing and Measurement

17.2 Collection

17.3 Analysis and Computation

17.4 Alerting and Escalation Manager

17.5 Visualization

17.6 Storage

17.7 Configuration

17.8 Summary

Exercises

## **18 Capacity Planning**



# **Table of Contents**

18.1 Standard Capacity Planning

18.2 Advanced Capacity Planning

18.3 Resource Regression

18.4 Launching New Services

18.5 Reduce Provisioning Time

18.6 Summary

Exercises

## **19 Creating KPIs**

19.1 What Is a KPI?

19.2 Creating KPIs

19.3 Example KPI: Machine Allocation

19.4 Case Study: Error Budget

19.5 Summary

Exercises

## **20 Operational Excellence**

20.1 What Does Operational Excellence Look Like?

20.2 How to Measure Greatness

20.3 Assessment Methodology

20.4 Service Assessments

20.5 Organizational Assessments

20.6 Levels of Improvement

20.7 Getting Started

20.8 Summary

Exercises

## **Epilogue**

## **Part III: Appendices**

### **A: Assessments**

A.1 Regular Tasks (RT)

A.2 Emergency Response (ER)

A.3 Monitoring and Metrics (MM)

# **Table of Contents**

- A.4 Capacity Planning (CP)
- A.5 Change Management (CM)
- A.6 New Product Introduction and Removal (NPI/NPR)
- A.7 Service Deployment and Decommissioning (SDD)
- A.8 Performance and Efficiency (PE)
- A.9 Service Delivery: The Build Phase
- A.10 Service Delivery: The Deployment Phase
- A.11 Toil Reduction
- A.12 Disaster Preparedness

## **B: The Origins and Future of Distributed Computing and Clouds**

- B.1 The Pre-Web Era (1985-1994)
  - Availability Requirements
  - Technology
  - Scaling
  - High Availability
  - Costs
- B.2 The First Web Era: The Bubble (1995-2000)
  - Availability Requirements
  - Technology
  - Scaling
  - High Availability
  - Costs
- B.3 The Dot-Bomb Era (2000-2003)
  - Availability Requirements
  - Technology
  - High Availability
  - Scaling
  - Costs
- B.4 The Second Web Era (2003-2010)
  - Availability Requirements
  - Technology

# **Table of Contents**

High Availability

Scaling

Costs

## **B.5 The Cloud Computing Era (2010present)**

Availability Requirements

Costs

Scaling and High Availability

Technology

## **B.6 Conclusion**

Exercises

## **C: Scaling Terminology and Concepts**

C.1 Constant, Linear, and Exponential Scaling

C.2 Big O Notation

C.3 Limitations of Big O Notation

## **D: Templates and Examples**

D.1 Design Document Template

D.2 Design Document Example

D.3 Sample Postmortem Template

## **E: Recommended Reading**

**Bibliography**

**Index**