

ANTHONY SEQUEIRA

Cert Guide

Learn, prepare, and practice for exam success



AWS Certified Solutions Architect - Associate (SAA-C01)

PEARSON IT
CERTIFICATION

Contents

1. Cover Page
2. About This E-Book
3. Title Page
4. Copyright Page
5. Contents at a Glance
6. Table of Contents
7. About the Author
8. Dedication
9. Acknowledgments
10. About the Technical Reviewer
11. We Want to Hear from You!
12. Reader Services
13. Introduction
14. Part I: Domain 1: Design Resilient Architectures

1. Chapter 1 The Fundamentals of AWS

1. “Do I Know This Already?” Quiz
2. Advantages of Cloud Technologies
3. An Overview of Key AWS Services
4. Review All Key Topics
5. Complete Tables and Lists from Memory
6. Define Key Terms
7. Q&A

2. Chapter 2 Designing Resilient Storage

1. “Do I Know This Already?” Quiz
2. Designing Resilient S3 Services
3. Designing Resilient EBS Services
4. Designing Resilient EFS Services
5. Designing Resilient Glacier Services
6. Review All Key Topics

7. Complete Tables and Lists from Memory
8. Define Key Terms
9. Q&A

3. Chapter 3 Designing Decoupling Mechanisms

1. “Do I Know This Already?” Quiz
2. Decoupling Demystified
3. Advantages of Decoupled Designs
4. Synchronous Decoupling
5. Asynchronous Decoupling
6. Review All Key Topics
7. Complete Tables and Lists from Memory
8. Define Key Terms
9. Q&A

4. Chapter 4 Designing a Multitier Infrastructure

1. “Do I Know This Already?” Quiz
2. Single-Tier Architectures
3. Multitier Architectures
4. The Classic Three-Tier Architecture
5. Review All Key Topics
6. Complete Tables and Lists from Memory
7. Define Key Terms
8. Q&A

5. Chapter 5 Designing High Availability Architectures

1. “Do I Know This Already?” Quiz
2. High Availability Compute
3. High Availability Application Services
4. High Availability Database Services
5. High Availability Networking Services
6. High Availability Storage Services
7. High Availability Security Services
8. High Availability Monitoring Services
9. Review All Key Topics
10. Complete Tables and Lists from Memory

11. Define Key Terms
 12. Q&A
15. Part II: Domain 2: Define Performant Architectures
1. Chapter 6 Choosing Performant Storage
 1. “Do I Know This Already?” Quiz
 2. Performant S3 Services
 3. Performant EBS Services
 4. Performant EFS Services
 5. Performant Glacier Services
 6. Review All Key Topics
 7. Complete Tables and Lists from Memory
 8. Define Key Terms
 9. Q&A
 2. Chapter 7 Choosing Performant Databases
 1. “Do I Know This Already?” Quiz
 2. Aurora
 3. RDS
 4. DynamoDB
 5. ElastiCache
 6. Redshift
 7. Review All Key Topics
 8. Complete Tables and Lists from Memory
 9. Define Key Terms
 10. Q&A
 3. Chapter 8 Improving Performance with Caching
 1. “Do I Know This Already?” Quiz
 2. ElastiCache
 3. DynamoDB Accelerator
 4. CloudFront
 5. Greengrass
 6. Route 53
 7. Review All Key Topics

8. Complete Tables and Lists from Memory
9. Define Key Terms
10. Q&A

4. Chapter 9 Designing for Elasticity

1. “Do I Know This Already?” Quiz
2. Elastic Load Balancing
3. Auto Scaling
4. Review All Key Topics
5. Complete Tables and Lists from Memory
6. Define Key Terms
7. Q&A

16. Part III: Domain 3: Specify Secure Applications and Architectures

1. Chapter 10 Securing Application Tiers

1. “Do I Know This Already?” Quiz
2. Using IAM
3. Securing the OS and Applications
4. Review All Key Topics
5. Complete Tables and Lists from Memory
6. Define Key Terms
7. Q&A

2. Chapter 11 Securing Data

1. “Do I Know This Already?” Quiz
2. Resource Access Authorization
3. Storing and Managing Encryption Keys in the Cloud
4. Protecting Data at Rest
5. Decommissioning Data Securely
6. Protecting Data in Transit
7. Review All Key Topics
8. Complete Tables and Lists from Memory
9. Define Key Terms
10. Q&A

3. Chapter 12 Networking Infrastructure for a Single VPC Application

1. “Do I Know This Already?” Quiz
2. Introducing the Basic AWS Network Infrastructure
3. Network Interfaces
4. Route Tables
5. Internet Gateways
6. Egress-Only Internet Gateways
7. DHCP Option Sets
8. DNS
9. Elastic IP Addresses
10. VPC Endpoints
11. NAT
12. VPC Peering
13. ClassicLink
14. Review All Key Topics
15. Complete Tables and Lists from Memory
16. Define Key Terms
17. Q&A

17. Part IV: Domain 4: Design Cost-Optimized Architectures

1. Chapter 13 Cost-Optimized Storage

1. “Do I Know This Already?” Quiz
2. S3 Services
3. EBS Services
4. EFS Services
5. Glacier Services
6. Review All Key Topics
7. Complete Tables and Lists from Memory
8. Define Key Terms
9. Q&A

2. Chapter 14 Cost-Optimized Compute

1. “Do I Know This Already?” Quiz
2. Cost-Optimized EC2 Services
3. Cost-Optimized Lambda Services
4. Review All Key Topics

5. Complete Tables and Lists from Memory
6. Define Key Terms
7. Q&A

18. Part V: Domain 5: Define Operationally Excellent Architectures

1. Chapter 15 Features for Operational Excellence

1. “Do I Know This Already?” Quiz
2. Introduction to the AWS Well-Architected Framework
3. Prepare
4. Operate
5. Evolve
6. Review All Key Topics
7. Complete Tables and Lists from Memory
8. Define Key Terms
9. Q&A

19. Part VI: Final Preparation

1. Chapter 16 Final Preparation

1. Exam Information
2. Getting Ready
3. Tools for Final Preparation
4. Suggested Plan for Final Review/Study
5. Summary

20. Part VII: Appendixes

1. Glossary
2. Appendix A Answers to the “Do I Know This Already?” Quizzes and Q&A Sections

1. Chapter 1
2. Chapter 2
3. Chapter 3
4. Chapter 4
5. Chapter 5

6. [Chapter 6](#)
7. [Chapter 7](#)
8. [Chapter 8](#)
9. [Chapter 9](#)
10. [Chapter 10](#)
11. [Chapter 11](#)
12. [Chapter 12](#)
13. [Chapter 13](#)
14. [Chapter 14](#)
15. [Chapter 15](#)

3. [Appendix B AWS Certified Solutions Architect – Associate \(SAA-C01\) Cert Guide Exam Updates](#)

1. [Always Get the Latest at the Book’s Product Page](#)
2. [Technical Content](#)

4. [Index](#)

21. [Online Elements](#)

1. [Glossary](#)
2. [Appendix C Memory Tables](#)

1. [Chapter 2](#)
2. [Chapter 4](#)
3. [Chapter 5](#)
4. [Chapter 6](#)
5. [Chapter 7](#)
6. [Chapter 11](#)

3. [Appendix D Memory Tables Answer Key](#)

1. [Chapter 2](#)
2. [Chapter 4](#)
3. [Chapter 5](#)
4. [Chapter 6](#)
5. [Chapter 7](#)
6. [Chapter 11](#)

4. Appendix E Study Planner

1. i
2. ii
3. iii
4. iv
5. v
6. vi
7. vii
8. viii
9. ix
10. x
11. xi
12. xii
13. xiii
14. xiv
15. xv
16. xvi
17. xvii
18. xviii
19. xix
20. xx
21. xxi
22. xxii
23. xxiii
24. xxiv
25. xxv
26. xxvi
27. xxvii
28. xxviii
29. xxix
30. 2
31. 3
32. 4
33. 5
34. 6
35. 7
36. 8
37. 9
38. 10

- 39. 11
- 40. 12
- 41. 13
- 42. 14
- 43. 15
- 44. 16
- 45. 17
- 46. 18
- 47. 19
- 48. 20
- 49. 21
- 50. 22
- 51. 23
- 52. 24
- 53. 25
- 54. 26
- 55. 27
- 56. 28
- 57. 29
- 58. 30
- 59. 31
- 60. 32
- 61. 33
- 62. 34
- 63. 35
- 64. 36
- 65. 37
- 66. 38
- 67. 39
- 68. 40
- 69. 41
- 70. 42
- 71. 43
- 72. 44
- 73. 45
- 74. 46
- 75. 47
- 76. 48
- 77. 49
- 78. 50

79.	51
80.	52
81.	53
82.	54
83.	55
84.	56
85.	57
86.	58
87.	59
88.	60
89.	61
90.	62
91.	63
92.	64
93.	65
94.	66
95.	67
96.	68
97.	69
98.	70
99.	71
100.	72
101.	73
102.	74
103.	75
104.	76
105.	77
106.	78
107.	79
108.	80
109.	81
110.	82
111.	83
112.	84
113.	85
114.	86
115.	87
116.	88
117.	89
118.	90

- 119. 91
- 120. 92
- 121. 93
- 122. 94
- 123. 95
- 124. 96
- 125. 97
- 126. 98
- 127. 99
- 128. 100
- 129. 101
- 130. 102
- 131. 103
- 132. 104
- 133. 105
- 134. 106
- 135. 107
- 136. 108
- 137. 109
- 138. 110
- 139. 111
- 140. 112
- 141. 113
- 142. 114
- 143. 115
- 144. 116
- 145. 117
- 146. 118
- 147. 119
- 148. 120
- 149. 121
- 150. 122
- 151. 123
- 152. 124
- 153. 125
- 154. 126
- 155. 127
- 156. 128
- 157. 129
- 158. 130

- 159. 131
- 160. 132
- 161. 133
- 162. 134
- 163. 135
- 164. 136
- 165. 137
- 166. 138
- 167. 139
- 168. 140
- 169. 141
- 170. 142
- 171. 143
- 172. 144
- 173. 145
- 174. 146
- 175. 147
- 176. 148
- 177. 149
- 178. 150
- 179. 151
- 180. 152
- 181. 153
- 182. 154
- 183. 155
- 184. 156
- 185. 157
- 186. 158
- 187. 159
- 188. 160
- 189. 161
- 190. 162
- 191. 163
- 192. 164
- 193. 165
- 194. 166
- 195. 167
- 196. 168
- 197. 169
- 198. 170

- 199. 171
- 200. 172
- 201. 173
- 202. 174
- 203. 175
- 204. 176
- 205. 177
- 206. 178
- 207. 179
- 208. 180
- 209. 181
- 210. 182
- 211. 183
- 212. 184
- 213. 185
- 214. 186
- 215. 187
- 216. 188
- 217. 189
- 218. 190
- 219. 191
- 220. 192
- 221. 193
- 222. 194
- 223. 195
- 224. 196
- 225. 197
- 226. 198
- 227. 199
- 228. 200
- 229. 201
- 230. 202
- 231. 203
- 232. 204
- 233. 205
- 234. 206
- 235. 207
- 236. 208
- 237. 209
- 238. 210

- 239. 211
- 240. 212
- 241. 213
- 242. 214
- 243. 215
- 244. 216
- 245. 217
- 246. 218
- 247. 219
- 248. 220
- 249. 221
- 250. 222
- 251. 223
- 252. 224
- 253. 225
- 254. 226
- 255. 227
- 256. 228
- 257. 229
- 258. 230
- 259. 231
- 260. 232
- 261. 233
- 262. 234
- 263. 235
- 264. 236
- 265. 237
- 266. 238
- 267. 239
- 268. 240
- 269. 241
- 270. 242
- 271. 243
- 272. 244
- 273. 245
- 274. 246
- 275. 247
- 276. 248
- 277. 249
- 278. 250

- 279. 251
- 280. 252
- 281. 253
- 282. 254
- 283. 255
- 284. 256
- 285. 257
- 286. 258
- 287. 259
- 288. 260
- 289. 261
- 290. 262
- 291. 263
- 292. 264
- 293. 265
- 294. 266
- 295. 267
- 296. 268
- 297. 269
- 298. 270
- 299. 271
- 300. 272
- 301. 273
- 302. 274
- 303. 275
- 304. 276
- 305. 277
- 306. 278
- 307. 279
- 308. 280
- 309. 281
- 310. 282
- 311. 283
- 312. 284
- 313. 285
- 314. 286
- 315. 287
- 316. 288
- 317. 289
- 318. 290

- 319. 291
- 320. 292
- 321. OG-1
- 322. OG-2
- 323. OG-3
- 324. OG-4
- 325. OG-5
- 326. OG-6
- 327. OG-7
- 328. OG-8
- 329. OC-1
- 330. OC-2
- 331. OC-3
- 332. OC-4
- 333. OD-1
- 334. OD-2
- 335. OD-3
- 336. OD-4

About This E-Book

EPUB is an open, industry-standard format for e-books. However, support for EPUB and its many features varies across reading devices and applications. Use your device or app settings to customize the presentation to your liking. Settings that you can customize often include font, font size, single or double column, landscape or portrait mode, and figures that you can click or tap to enlarge. For additional information about the settings and features on your reading device or app, visit the device manufacturer's Web site.

Many titles include programming code or configuration examples. To optimize the presentation of these elements, view the e-book in single-column, landscape mode and adjust the font size to the smallest setting. In addition to presenting code and configurations in the reflowable text format, we have included images of the code that mimic the presentation found in the print book; therefore, where the reflowable format may compromise the presentation of the code listing, you will see a “Click here to view code image” link. Click the link to view the print-fidelity code image. To return to the previous page viewed, click the Back button on your device or app.

AWS Certified Solutions Architect - Associate (SAA- C01) Cert Guide

Anthony Sequeira, CCIE No. 15626



AWS Certified Solutions Architect - Associate (SAA-C01) Cert Guide

Copyright © 2019 by Pearson Education, Inc.

All rights reserved. No part of this book shall be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise, without written permission from the publisher. No patent liability is assumed with respect to the use of the information contained herein. Although every precaution has been taken in the preparation of this book, the publisher and author assume no responsibility for errors or omissions. Nor is any liability assumed for damages resulting from the use of the information contained herein.

ISBN-13: 978-0-7897-6049-4

ISBN-10: 0-7897-6049-5

Library of Congress Control Number: 2018963110

01 19

Trademarks

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Pearson IT Certification cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

Warning and Disclaimer

Every effort has been made to make this book as complete and

as accurate as possible, but no warranty or fitness is implied. The information provided is on an “as is” basis. The author and the publisher shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book.

Special Sales

For information about buying this title in bulk quantities, or for special sales opportunities (which may include electronic versions; custom cover designs; and content particular to your business, training goals, marketing focus, or branding interests), please contact our corporate sales department at corpsales@pearsoned.com or (800) 382-3419.

For government sales inquiries, please contact governmentsales@pearsoned.com.

For questions about sales outside the U.S., please contact intlcs@pearson.com.

Editor-in-Chief

Mark Taub

Product Line Manager

Brett Bartow

Acquisitions Editor

Paul Carlstroem

Development Editor

Christopher Cleveland

Managing Editor

Sandra Schroeder

Senior Project Editor

Tonya Simpson

Copy Editor

Chuck Hutchinson

Indexer

Erika Millen

Proofreader

Abigail Manheim

Technical Editor

Ryan Dymek

Publishing Coordinator

Cindy J. Teeters

Cover Designer

Chuti Prasertsith

Composer

codemantra

Contents at a Glance

Introduction

Part I: Domain 1: Design Resilient Architectures

CHAPTER 1 The Fundamentals of AWS

CHAPTER 2 Designing Resilient Storage

CHAPTER 3 Designing Decoupling Mechanisms

CHAPTER 4 Designing a Multitier Infrastructure

CHAPTER 5 Designing High Availability Architectures

Part II: Domain 2: Define Performant Architectures

CHAPTER 6 Choosing Performant Storage

CHAPTER 7 Choosing Performant Databases

CHAPTER 8 Improving Performance with Caching

CHAPTER 9 Designing for Elasticity

Part III: Domain 3: Specify Secure Applications and Architectures

CHAPTER 10 Securing Application Tiers

CHAPTER 11 Securing Data

CHAPTER 12 Networking Infrastructure for a Single VPC Application

Part IV: Domain 4: Design Cost-Optimized Architectures

CHAPTER 13 Cost-Optimized Storage

CHAPTER 14 Cost-Optimized Compute

Part V: Domain 5: Define Operationally Excellent Architectures

CHAPTER 15 Features for Operational Excellence

Part VI: Final Preparation

CHAPTER 16 Final Preparation

Part VII: Appendixes

Glossary

APPENDIX A Answers to the “Do I Know This Already?”

Quizzes and Q&A Sections

APPENDIX B AWS Certified Solutions Architect – Associate (SAA-C01) Cert Guide Exam Updates

Index

Online Elements

Glossary

APPENDIX C Memory Tables

APPENDIX D Memory Tables Answer Key

APPENDIX E Study Planner

Table of Contents

Introduction

Part I: Domain 1: Design Resilient Architectures

Chapter 1 The Fundamentals of AWS

“Do I Know This Already?” Quiz

Advantages of Cloud Technologies

An Overview of Key AWS Services

Compute Services

Elastic Compute Cloud

Lambda

Elastic Container Service

Elastic Container Registry

Elastic Container Service for Kubernetes

Fargate

Serverless Application Repository

Lightsail

AWS Batch

Elastic Beanstalk

Elastic Load Balancing

Auto Scaling

CloudFormation

Application Services

OpsWorks

CloudFront

Simple Queue Service

Simple Notification Service

Kinesis

Database Services

Aurora

Relational Database Service

DynamoDB

ElastiCache

Redshift

Database Migration Service

Networking Services

The AWS Global Infrastructure

Virtual Private Cloud

Direct Connect

Route 53

Storage Services

Simple Storage Service

Elastic Block Store

Elastic File System

Glacier

Snowball

AWS Storage Gateway

Security Services

Identity and Access Management

Web Application Firewall

Key Management Service

Directory Services

Management Services

Trusted Advisor

CloudWatch

CloudTrail

Review All Key Topics

Complete Tables and Lists from Memory

Define Key Terms

Q&A

Chapter 2 Designing Resilient Storage

“Do I Know This Already?” Quiz

Designing Resilient S3 Services

S3 Storage Classes

Lab: Creating an S3 Bucket

Lab Cleanup

Designing Resilient EBS Services

EBS Versus Instance Stores

Elastic Block Store

Lab: Creating an EBS Volume

Lab Cleanup

Elastic Block Store

Designing Resilient EFS Services

Lab: A Basic EFS Configuration

Lab Cleanup

Designing Resilient Glacier Services

Lab: Creating a Vault

Lab Cleanup

Review All Key Topics

Complete Tables and Lists from Memory

Define Key Terms

Q&A

Chapter 3 Designing Decoupling Mechanisms

“Do I Know This Already?” Quiz

Decoupling Demystified

Advantages of Decoupled Designs

Synchronous Decoupling

Asynchronous Decoupling

Lab: Configure SQS

Lab Cleanup

Review All Key Topics

Complete Tables and Lists from Memory

Define Key Terms

Q&A

Chapter 4 Designing a Multitier Infrastructure

“Do I Know This Already?” Quiz

Single-Tier Architectures

Lab: Building a Single-Tier Architecture with

EC2

Lab Cleanup

Multitier Architectures

The Classic Three-Tier Architecture

Review All Key Topics

Complete Tables and Lists from Memory

Define Key Terms

Q&A

Chapter 5 Designing High Availability Architectures

“Do I Know This Already?” Quiz

High Availability Compute

Lab: Provisioning EC2 Instances in Different Availability Zones

Lab Cleanup

High Availability Application Services

High Availability Database Services

High Availability Networking Services

High Availability Storage Services

High Availability Security Services

High Availability Monitoring Services

Review All Key Topics

Complete Tables and Lists from Memory

Define Key Terms

Q&A

Part II: Domain 2: Define Performant Architectures

Chapter 6 Choosing Performant Storage

“Do I Know This Already?” Quiz

Performant S3 Services

Performant EBS Services

Performant EFS Services

Performant Glacier Services

Review All Key Topics

Complete Tables and Lists from Memory

Define Key Terms

Q&A

Chapter 7 Choosing Performant Databases

“Do I Know This Already?” Quiz

Aurora

Which DB Instance Are You Connected To?

When to Use T2 Instances

Work with Asynchronous Key Prefetch

Avoid Multithreaded Replication

Use Scale Reads

Consider Hash Joins

Use TCP Keepalive Parameters

RDS

DynamoDB

Burst Capacity

Adaptive Capacity

Secondary Indexes

Querying and Scanning Data

ElastiCache

Lazy Loading Versus Write Through

Scenario 1: Cache Hit

Scenario 2: Cache Miss

Advantages and Disadvantages of Lazy Loading

Advantages and Disadvantages of Write Through

What Is TTL?

Background Write Process and Memory Usage

Avoid Running Out of Memory When Executing
a Background Write

How Much Reserved Memory Do You Need?

Parameters to Manage Reserved Memory

Online Cluster Resizing

Redshift

Amazon Redshift Best Practices for Designing
Queries

Work with Recommendations from Amazon
Redshift Advisor

Review All Key Topics

Complete Tables and Lists from Memory

Define Key Terms

Q&A

Chapter 8 Improving Performance with Caching

“Do I Know This Already?” Quiz

ElastiCache

Lab: Configuring a Redis ElastiCache Cluster

Lab Cleanup

DynamoDB Accelerator

CloudFront

Lab: Configuring CloudFront

Lab Cleanup

Greengrass

Route 53

Lab: Creating a Hosted Domain and DNS

Records in Route 53

Lab Cleanup

Review All Key Topics

Complete Tables and Lists from Memory

Define Key Terms

Q&A

Chapter 9 Designing for Elasticity

“Do I Know This Already?” Quiz

Elastic Load Balancing

Auto Scaling

Target Tracking Scaling Policies

The Cooldown Period

Review All Key Topics

Complete Tables and Lists from Memory

Define Key Terms

Q&A

Part III: Domain 3: Specify Secure Applications and Architectures

Chapter 10 Securing Application Tiers

“Do I Know This Already?” Quiz

Using IAM

IAM Identities

Securing the OS and Applications

Security Groups

Network ACLs

Systems Manager Patch Manager

Review All Key Topics

Complete Tables and Lists from Memory

Define Key Terms

Q&A

Chapter 11 Securing Data

“Do I Know This Already?” Quiz

Resource Access Authorization

Storing and Managing Encryption Keys in the Cloud

Protecting Data at Rest

Decommissioning Data Securely

Protecting Data in Transit

Review All Key Topics

Complete Tables and Lists from Memory

Define Key Terms

Q&A

Chapter 12 Networking Infrastructure for a Single VPC Application

“Do I Know This Already?” Quiz

Introducing the Basic AWS Network Infrastructure

Lab: Checking Default Networking Components in a Region

Network Interfaces

Route Tables

Internet Gateways

Egress-Only Internet Gateways

DHCP Option Sets

DNS

Elastic IP Addresses

VPC Endpoints

Interface Endpoints (Powered by AWS PrivateLink)

Gateway Endpoints

NAT

VPC Peering

ClassicLink

Review All Key Topics

Complete Tables and Lists from Memory

Define Key Terms

Q&A

Part IV: Domain 4: Design Cost-Optimized Architectures

Chapter 13 Cost-Optimized Storage

“Do I Know This Already?” Quiz

S3 Services

Lab: Estimating AWS S3 Costs

Lab: Implementing Lifecycle Management

Lab Cleanup

EBS Services

EFS Services

Glacier Services

Lab: Changing the Retrieval Rate in Glacier

Lab Cleanup

Review All Key Topics

Complete Tables and Lists from Memory

Define Key Terms

Q&A

Chapter 14 Cost-Optimized Compute

“Do I Know This Already?” Quiz

Cost-Optimized EC2 Services

Lab: Using the Cost Explorer

Lab: Creating a Billing Alarm

Cost-Optimized Lambda Services

Review All Key Topics

Complete Tables and Lists from Memory

Define Key Terms

Q&A

Part V: Domain 5: Define Operationally Excellent Architectures

Chapter 15 Features for Operational Excellence

“Do I Know This Already?” Quiz

Introduction to the AWS Well-Architected Framework

Prepare

Operational Priorities

Design for Operations

Operational Readiness

Operate

Understanding Operational Health

Responding to Events

Evolve

Learning from Experience

Share Learnings

Review All Key Topics

Complete Tables and Lists from Memory

Define Key Terms

Q&A

Part VI: Final Preparation

Chapter 16 Final Preparation

Exam Information

Getting Ready

Tools for Final Preparation

Pearson Test Prep Practice Test Engine and Questions on the Website

Accessing the Pearson Test Prep Practice Test Software Online

Accessing the Pearson Test Prep Practice Test Software Offline

Customizing Your Exams

Updating Your Exams

Premium Edition

Memory Tables

Chapter-Ending Review Tools

Suggested Plan for Final Review/Study

Summary

Part VII: Appendixes

Glossary

Appendix A Answers to the “Do I Know This Already?”

Quizzes and Q&A Sections

Appendix B AWS Certified Solutions Architect – Associate (SAA-C01) Cert Guide Exam Updates

Index

Online Elements

Glossary

Appendix C Memory Tables

Appendix D Memory Tables Answer Key

Appendix E Study Planner

About the Author

Anthony Sequeira, CCIE No. 15626, is a seasoned trainer and author regarding various levels and tracks of Cisco, Microsoft, and AWS certifications. In 1994, Anthony formally began his career in the information technology industry with IBM in Tampa, Florida. He quickly formed his own computer consultancy, Computer Solutions, and then discovered his true passion—teaching and writing about information technologies.

Anthony joined Mastering Computers in 1996 and lectured to massive audiences around the world about the latest in computer technologies. Mastering Computers became the revolutionary online training company KnowledgeNet, and Anthony trained there for many years.

Anthony is currently pursuing his second CCIE in the area of Cisco Data Center! He is a full-time instructor at CBT Nuggets.

Dedication

*I dedicate this book to my incredible daughter, Bella Sequeira.
While she may never read it to master AWS, she can at least
use it as a rather expensive coaster in her beautiful new
Oregon home.*

Acknowledgments

This manuscript was made truly great by the incredible technical review of Ryan Dymek. Sometimes I think he might have invented AWS.

I would also like to express my gratitude to Chris Cleveland, development editor of this book. I was so incredibly lucky to work with him again on this text. Like Ryan, he made this book several cuts above the rest.

About the Technical Reviewer

Ryan Dymek has been working with Amazon Web Services (AWS) for more than 9 years and holds all nine AWS certifications as well as various Google Cloud Platform (GCP) certifications. Ryan trains and advises some of the largest companies in the world on sound architectural practices in cloud strategy and DevOps principles. While working with business leaders, developers, and engineers, Ryan bridges the gap between business and technology, maintaining the understanding and skills required to be able to perform at a deep technical level. Ryan runs his own cloud consulting practice advising more than 20 companies on the Fortune 500 list and has helped many startups find their way in the cloud.

In addition to cloud and technical acumen, Ryan is a certified business coach personally trained by John Maxwell. He uses these professional skills not only to advise companies on best cloud practices but also on how to align with a business's needs and culture, making confident business and technical decisions and cultivating a transformation into DevOps.

We Want to Hear from You!

As the reader of this book, *you* are our most important critic and commentator. We value your opinion and want to know what we're doing right, what we could do better, what areas you'd like to see us publish in, and any other words of wisdom you're willing to pass our way.

We welcome your comments. You can email to let us know what you did or didn't like about this book—as well as what we can do to make our books better.

Please note that we cannot help you with technical problems related to the topic of this book.

When you write, please be sure to include this book's title and author as well as your name and email address. We will carefully review your comments and share them with the author and editors who worked on the book.

Email: feedback@pearsonitcertification.com

Reader Services

Register your copy of *AWS Certified Solutions Architect - Associate (SAA-C01) Cert Guide* at www.pearsonitcertification.com for convenient access to downloads, updates, and corrections as they become available. To start the registration process, go to www.pearsonitcertification.com/register and log in or create an account.* Enter the product ISBN 9780789760494 and click Submit. When the process is complete, you will find any available bonus content under Registered Products.

*Be sure to check the box that you would like to hear from us to receive exclusive discounts on future editions of this product.

Introduction

The AWS Certified Solutions Architect - Associate is a cloud-related certification that tests a candidate's ability to architect effective solutions calling upon the most popular aspects of Amazon Web Services. The solutions architect candidates must demonstrate their skills on how to plan and implement a sophisticated design that saves costs, is secure, and perhaps most importantly, operates with excellence. Candidates are also required to know the most important facts regarding various services and their capabilities.

The AWS Certified Solutions Architect is an Associate-level cloud career certification. This certification is an excellent second step after the achievement of the AWS Certified Cloud Practitioner certification, although seasoned AWS users might choose to skip that entry-level exam. Following this certification, AWS offers a Professional level of certification for the solutions architect.

AWS also offers certifications in which you might be interested in different tracks. For example, a Developer track for AWS also includes Associate and Professional levels. Amazon also uses specialty certifications to deep dive into many different areas such as security and advanced networking.

Note

The AWS Certified Solutions Architect - Associate certification is globally recognized and does an excellent job of demonstrating that the holder has knowledge and skills across a broad range of AWS topics.

THE GOALS OF THE AWS CERTIFIED SOLUTIONS ARCHITECT - ASSOCIATE CERTIFICATION

The AWS Certified Solutions Architect - Associate certification is intended for individuals who perform a solutions architect role. The certification seeks to validate your ability to effectively demonstrate knowledge of how to architect and deploy secure and robust applications on AWS technologies.

You should be able to define a solution using architectural design principles based on customer requirements. You should also be able to provide implementation guidance based on best practices to the organization throughout the lifecycle of the project.

Ideal Candidates

Although this text provides you with the information required to pass this exam, Amazon considers ideal candidates to be those who possess the following:

- ■ One year of hands-on experience designing available, cost-efficient, fault-tolerant, and scalable distributed systems on AWS
- ■ Hands-on experience using computing, networking, storage, and database AWS services
- ■ Hands-on experience with AWS deployment and management services
- ■ The ability to identify and define technical requirements for an AWS-based application
- ■ The ability to identify which AWS services meet a given technical requirement
- ■ Knowledge of recommended best practices for building secure and

reliable applications on the AWS platform

- ■ An understanding of the basic architectural principles of building on the AWS cloud
- ■ An understanding of the AWS global infrastructure
- ■ An understanding of network technologies as they relate to AWS
- ■ An understanding of security features and tools that AWS provides and how they relate to traditional services

THE EXAM OBJECTIVES (DOMAINS)

The AWS Certified Solutions Architect - Associate (SAA-C01) exam is broken down into five major domains. The contents of this book cover each of the domains and the subtopics included in them as illustrated in the following descriptions.

The following table breaks down each domain represented in the exam.

Domain	Percentage of Representation in Exam
1: Design Resilient Architectures	34%
2: Define Performant Architectures	24%
3: Specify Secure Applications and Architectures	26%

4: Design Cost-Optimized Architectures	10%
5: Define Operationally Excellent Architectures	6%
Total 100%	

1.0 Design Resilient Architectures

The Design Resilient Architectures domain is covered in Chapters 1 through 5. It covers critical information for designing and deploying highly available services and resources in the cloud. It comprises 34 percent of the exam. Topics include

- ■ 1.1 Choose reliable/resilient storage.
- ■ 1.2 Determine how to design decoupling mechanisms using AWS services.
- ■ 1.3 Determine how to design a multitier architecture solution.
- ■ 1.4 Determine how to design high availability and/or fault-tolerant architectures.

2.0 Define Performant Architectures

The Define Performant Architectures domain is covered in Chapters 6 through 9. This domain ensures that you consider and properly implement solutions that perform per the client requirements. This makes up 24 percent of the exam. Topics include

- ■ 2.1 Choose performant storage and databases.

- ■ 2.2 Apply caching to improve performance.
- ■ 2.3 Design solutions for elasticity and scalability.

3.0 Specify Secure Applications and Architectures

The Specify Secure Applications and Architectures domain is covered in Chapters 10 through 12. This domain is critical, especially in today’s landscape, because it assists you in an AWS implementation that features security at many layers. This builds a Defense in Depth type of strategy that increases your chances of warding off would-be attackers. It encompasses 26 percent of the exam. Topics include

- ■ 3.1 Determine how to secure application tiers.
- ■ 3.2 Determine how to secure data.
- ■ 3.3 Define the networking infrastructure for a single VPC application.

4.0 Design Cost-Optimized Architectures

The Design Cost-Optimized Architectures domain is covered in Chapters 13 and 14. Here you learn about saving on costs where these costs could be the most substantial—in the areas of compute and storage. This domain embodies 10 percent of the exam. The topics include

- ■ 4.1 Determine how to design cost-optimized storage.
- ■ 4.2 Determine how to design cost-optimized compute.

5.0 Define Operationally Excellent Architectures

The Define Operationally Excellent Architectures domain is covered in Chapter 15. This chapter ensures you understand how to use AWS best practices around the three main steps of Prepare, Operate, and Evolve with AWS solutions. This domain makes up 6 percent of the exam. Topics include

- ■ 5.1 Choose design features in solutions that enable operational excellence.

STEPS TO BECOMING AN AWS CERTIFIED SOLUTIONS ARCHITECT – ASSOCIATE

To become an AWS Certified Solutions Architect - Associate, a test candidate should meet certain prerequisites (none of these are formal prerequisites) and follow specific procedures. Test candidates must qualify for the exam and sign up for the exam.

Recommended Experience

There are no prerequisites for the Solutions Architect - Associate certification. However, Amazon recommends that candidates possess the Certified Cloud Practitioner certification or equivalent knowledge.

Note

Other certifications you might possess in related areas, such as Microsoft Azure, can also prove beneficial.

Signing Up for the Exam

The steps required to sign up for the Solutions Architect - Associate exam are as follows:

Step 1. Create an AWS Certification account at

<https://www.aws.training/> Certification and schedule your exam.

Step 2. Complete the Examination Agreement, attesting to the truth of your assertions regarding professional experience and legally committing to the adherence of the testing policies.

Step 3. Submit the examination fee.

FACTS ABOUT THE EXAM

The exam is a computer-based test. The exam consists of multiple-choice questions only. You must bring a government-issued identification card. No other forms of ID will be accepted.

Tip

Refer to the AWS Certification site at <https://aws.amazon.com/certification/> for more information regarding this, and other, AWS Certifications. I am also in the process of building a simple hub site for everything AWS Certification related at awscerthub.com. This site is made up of 100 percent AWS solutions. Of course!

ABOUT THE SOLUTIONS ARCHITECT – ASSOCIATE CERT GUIDE

This book maps directly to the topic areas of the exam and uses a number of features to help you understand the topics and prepare for the exam.

Objectives and Methods

This book uses several key methodologies to help you discover the exam topics on which you need more review, to help you fully understand and remember those details, and to help you prove to yourself that you have retained your knowledge of those topics. This book does not try to help you pass the exam only by memorization; it seeks to help you to truly learn and understand the topics. This book is designed to help you pass the AWS Certified Solutions Architect - Associate exam by using the following methods:

- ■ Helping you discover which exam topics you have not mastered
- ■ Providing explanations and information to fill in your knowledge gaps
- ■ Supplying exercises that enhance your ability to recall and deduce the answers to test questions
- ■ Providing practice exercises on the topics and the testing process via test questions on the companion website

Book Features

To help you customize your study time using this book, the core chapters have several features that help you make the best use of your time:

- ■ **Foundation Topics:** These are the core sections of each chapter. They explain the concepts for the topics in that chapter.
- ■ **Exam Preparation Tasks:** After the “Foundation Topics” section of each chapter, the “Exam Preparation Tasks” section lists a series of study activities that you should do at the end of the chapter:
 - ■ **Review All Key Topics:** The Key Topic icon appears next to the most important items in the “Foundation Topics” section of the chapter. The “Review All Key Topics” activity lists the key topics from the chapter, along with their page numbers. Although the contents of the entire chapter could be on the exam, you should definitely know the information listed in each key topic, so you should review these.
 - ■ **Define Key Terms:** Although the Solutions Architect exam may be unlikely to ask a question such as “Define this term,” the exam does require that you learn and know a lot of new terminology. This section lists the most important terms from the chapter, asking you to write a short definition and compare your answer to the glossary at the end of the book.
 - ■ **Review Questions:** Confirm that you understand the content that you just covered by answering these questions and reading the answer explanations.
- ■ **Web-based Practice Exam:** The companion website includes the Pearson Cert Practice test engine that allows you to take practice exam questions. Use it to prepare with a sample exam and to pinpoint topics where you need more

study.

How This Book Is Organized

This book contains 15 core chapters—Chapters 1 through 15. Chapter 16 includes preparation tips and suggestions for how to approach the exam. The core chapters map to the AWS Certified Solutions Architect - Associate exam topic areas and cover the concepts and technologies that you will encounter on the exam.

COMPANION WEBSITE

Register this book to get access to the Pearson Test Prep practice test engine and other study materials plus additional bonus content. Check this site regularly for new and updated postings written by the authors that provide further insight into the more troublesome topics on the exam. Be sure to check the box that you would like to hear from us to receive updates and exclusive discounts on future editions of this product or related products.

To access this companion website, follow these steps:

Step 1. Go to www.pearsonITcertification.com/register and log in or create a new account.

Step 2. Enter the ISBN: **9780789760494**.

Step 3. Answer the challenge question as proof of purchase.

Step 4. Click the **Access Bonus Content** link in the Registered Products section of your account page, to be taken to the page where your downloadable content is available.

Please note that many of our companion content files can be very large, especially image and video files.

If you are unable to locate the files for this title by following these steps, please visit www.pearsonITcertification.com/contact and select the **Site Problems/Comments** option. Our customer service representatives will assist you.

PEARSON TEST PREP PRACTICE TEST SOFTWARE

As noted previously, this book comes complete with the Pearson Test Prep practice test software containing two full exams. These practice tests are available to you either online or as an offline Windows application. To access the practice exams that were developed with this book, please see the instructions in the card inserted in the sleeve in the back of the book. This card includes a unique access code that enables you to activate your exams in the Pearson Test Prep software.

ACCESSING THE PEARSON TEST PREP SOFTWARE ONLINE

The online version of this software can be used on any device with a browser and connectivity to the Internet, including desktop machines, tablets, and smartphones. To start using your practice exams online, follow these steps:

Step 1. Go to <https://www.PearsonTestPrep.com>.

Step 2. Select **Pearson IT Certification** as your product group.

Step 3. Enter your email/password for your account. If you

don't have an account on PearsonITCertification.com or CiscoPress.com, you will need to establish one by going to PearsonITCertification.com/join.

Step 4. In the My Products tab, click the **Activate New Product** button.

Step 5. Enter the access code printed on the insert card in the back of your book to activate your product.

Step 6. The product will now be listed in your My Products page. Click the **Exams** button to launch the exam settings screen and start your exam.

ACCESSING THE PEARSON TEST PREP SOFTWARE OFFLINE

If you wish to study offline, you can download and install the Windows version of the Pearson Test Prep software. You can find a download link for this software on the book's companion website, or you can just enter this link in your browser:

<http://www.pearsonitcertification.com/content/downloads/pcpt/engine.zip>

To access the book's companion website and the software, follow these steps:

Step 1. Register your book by going to PearsonITCertification.com/register and entering the ISBN: **9780789760494**.

Step 2. Answer the challenge questions.

Step 3. Go to your account page and click the **Registered Products** tab.

Step 4. Click the **Access Bonus Content** link under the product listing.

Step 5. Click the **Install Pearson Test Prep Desktop Version** link under the Practice Exams section of the page to download the software.

Step 6. After the software finishes downloading, unzip all the files on your computer.

Step 7. Double-click the application file to start the installation, and follow the onscreen instructions to complete the registration.

Step 8. After the installation is complete, launch the application and click the **Activate Exam** button on the My Products tab.

Step 9. Click the **Activate a Product** button in the Activate Product Wizard.

Step 10. Enter the unique access code found on the card in the sleeve in the back of your book and click the **Activate** button.

Step 11. Click **Next**, and then click **Finish** to download the exam data to your application.

Step 12. Start using the practice exams by selecting the product and clicking the **Open Exam** button to open the exam settings screen.

Note that the offline and online versions will synch together, so saved exams and grade results recorded on one version will

be available to you on the other as well.

CUSTOMIZING YOUR EXAMS

Once you are in the exam settings screen, you can choose to take exams in one of three modes:

- **Study mode:** Allows you to fully customize your exams and review answers as you are taking the exam. This is typically the mode you would use first to assess your knowledge and identify information gaps.
- **Practice Exam mode:** Locks certain customization options, as it is presenting a realistic exam experience. Use this mode when you are preparing to test your exam readiness.
- **Flash Card mode:** Strips out the answers and presents you with only the question stem. This mode is great for late-stage preparation when you really want to challenge yourself to provide answers without the benefit of seeing multiple-choice options. This mode does not provide the detailed score reports that the other two modes do, so you should not use it if you are trying to identify knowledge gaps.

In addition to these three modes, you will be able to select the source of your questions. You can choose to take exams that cover all of the chapters, or you can narrow your selection to just a single chapter or the chapters that make up specific parts in the book. All chapters are selected by default. If you want to narrow your focus to individual chapters, simply deselect all the chapters and then select only those on which you wish to focus in the Objectives area.

You can also select the exam banks on which to focus. Each exam bank comes complete with a full exam of questions that cover topics in every chapter. The two exams printed in the book are available to you as well as two additional exams of unique questions. You can have the test engine serve up exams from all four banks or just from one individual bank by selecting the desired banks in the exam bank area.

You can make several other customizations to your exam from the exam settings screen, such as the time of the exam, the number of questions served up, whether to randomize questions and answers, whether to show the number of correct answers for multiple-answer questions, and whether to serve up only specific types of questions. You can also create custom test banks by selecting only questions that you have marked or questions on which you have added notes.

UPDATING YOUR EXAMS

If you are using the online version of the Pearson Test Prep practice test software, you should always have access to the latest version of the software as well as the exam data. If you are using the Windows desktop version, every time you launch the software while connected to the Internet, it checks if there are any updates to your exam data and automatically downloads any changes that were made since the last time you used the software.

Sometimes, due to many factors, the exam data may not fully download when you activate your exam. If you find that figures or exhibits are missing, you may need to manually update your exams. To update a particular exam you have already activated and downloaded, simply click the **Tools** tab and click the **Update Products** button. Again, this is only an issue with the desktop Windows application.

If you wish to check for updates to the Pearson Test Prep exam engine software, Windows desktop version, simply click the **Tools** tab and click the **Update Application** button. This ensures that you are running the latest version of the software engine.

Figure Credits

Chapter 1, Figures 1-1 through 1-7, screenshot of AWS © 2018, Amazon Web Services, Inc.

Chapter 2, Figures 2-1 through 2-4, screenshot of AWS © 2018, Amazon Web Services, Inc.

Chapter 3, Figures 3-1 through 3-3, screenshot of AWS © 2018, Amazon Web Services, Inc.

Chapter 4, Figure 4-2, screenshot of AWS © 2018, Amazon Web Services, Inc.

Chapter 5, Figures 5-1 and 5-2, screenshot of AWS © 2018, Amazon Web Services, Inc.

Chapter 6, quote from Amazon in the note courtesy of Amazon Web Services, Inc.

Chapter 6, Figure 6-1, screenshot of AWS © 2018, Amazon Web Services, Inc.

Chapter 7, Figures 7-1 and 7-2, screenshot of AWS © 2018, Amazon Web Services, Inc.

Chapter 8, Figures 8-1 through 8-4, screenshot of AWS © 2018, Amazon Web Services, Inc.

Chapter 9, Figures 9-1 and 9-2, screenshot of AWS © 2018, Amazon Web Services, Inc.

Chapter 10, Figures 10-1 through 10-4, screenshot of AWS © 2018, Amazon Web Services, Inc.

Chapter 11, reference to NIST 800-88 courtesy of NIST 800-

88 “Guidelines for Media Sanitization,” Recommendations of the National Institute of Standards and Technology, Richard Kissel September, 2006.

Chapter 12, Figures 12-1 through 12-3, screenshot of AWS © 2018, Amazon Web Services, Inc.

Chapter 13, Figures 13-1 through 13-3, screenshot of AWS © 2018, Amazon Web Services, Inc.

Chapter 14, Figures 14-1 through 14-3, screenshot of AWS © 2018, Amazon Web Services, Inc.

Part I

Domain 1: Design Resilient Architectures

Chapter 1. The Fundamentals of AWS

This chapter covers the following subjects:

- ■ **Advantages of Cloud Technologies:** This section covers just some of the many advantages of cloud technologies in general. This section is not specific to Amazon Web Services (AWS), but AWS does provide all the benefits covered.
- ■ **An Overview of Key AWS Compute Services:** Compute services are a foundational part of AWS. This section examines the most important of these services that enable the creation of virtual machines, or even enable serverless computing in the public cloud.
- ■ **An Overview of Key AWS Application Services:** Application services enhance the capabilities of your cloud applications. For example, you might need to notify AWS administrators regarding essential application events. This section describes several of the more popular application services and what features they allow you to take advantage of.
- ■ **An Overview of Key AWS Database Services:** AWS is an excellent host for many different database technologies including relational, nonrelational, and data warehouse technologies. This section ensures you can recognize the different primary database services.
- ■ **An Overview of Key AWS Networking Services:** Understanding the components that make up the networks of AWS is essential. This section provides an overview of the critical network services.
- ■ **An Overview of Key AWS Storage Services:** AWS offers many different forms of data storage. This section describes these assorted options.
- ■ **An Overview of Key AWS Security Services:** AWS accommodates powerful security thanks to a wide variety of services. This section gives you an overview of the primary security services.
- ■ **An Overview of Key AWS Management Services:** You need robust management capabilities for the cloud to be useful for your organization. This section gives you an overview of the various management and

monitoring tools you can take advantage of with AWS.

This chapter has two primary goals. First, it ensures that you are familiar with some of the many advantages of cloud computing in general. Second, the chapter focuses on Amazon Web Services by introducing you to many of its essential services. These services are organized nicely for you by their crucial role in a cloud-based infrastructure. For example, the section for database services contains only services that relate to—you guessed it—databases of various types.

“DO I KNOW THIS ALREADY?” QUIZ

The “Do I Know This Already?” quiz allows you to assess whether you should read the entire chapter. Table 1-1 lists the major headings in this chapter and the “Do I Know This Already?” quiz questions covering the material in those headings so you can assess your knowledge of these specific areas. The answers to the “Do I Know This Already?” quiz appear in Appendix A.

Table 1-1 “Do I Know This Already?” Foundation Topics
Section-to-Question Mapping

Foundation Topics Section	Questions
Advantages of Cloud Technologies	1–3
An Overview of Key AWS Services	4–10

Caution

The goal of self-assessment is to gauge your mastery of the topics in this chapter. If you do not know the answer to a question or are only partially sure of the answer, you should mark that question as wrong for purposes of the self-assessment. Giving yourself credit for an answer you correctly guess skews your self-assessment results and might provide you with a false sense of security.

- 1.** Which would not be considered a business-related advantage to cloud technology?
 - a.** Flexibility
 - b.** “Pay as you go” model
 - c.** Lack of contractual commitments
 - d.** Increase in initial CapEx

- 2.** What is the type of security model often seen with public cloud?
 - a.** Shared responsibility model
 - b.** Hardware-based
 - c.** Sole responsibility model
 - d.** SLA-based model

- 3.** What is the primary feature that allows elasticity in the cloud?
 - a.** Route 53
 - b.** CapEx
 - c.** Auto scaling
 - d.** IAM

- 4.** What type of EC2 pricing model allows you to bid on unused AWS capacity?

- a.** Reserved instances
 - b.** On-demand instances
 - c.** Auction instances
 - d.** Spot instances
- 5.** What is the main serverless compute option in AWS?
 - a.** VPC
 - b.** Kinesis
 - c.** Lambda
 - d.** RDS
- 6.** You have a web application you have created with .NET with the intent to run it on an IIS server. What AWS service can handle the deployment for you?
 - a.** Lambda
 - b.** EC2
 - c.** Elastic Beanstalk
 - d.** Elastic Load Balancer
- 7.** Aurora is designed to offer what level of availability?
 - a.** 99.99%
 - b.** 99%
 - c.** 99.9999%
 - d.** 90%
- 8.** Which is not a valid engine choice with RDS?
 - a.** Aurora
 - b.** MySQL
 - c.** MongoDB

- d.** MariaDB
- 9.** What service would you use for scalable, fast, and flexible object-based storage for jpegs of various sizes as well as text files?
- a.** EFS
 - b.** EBS
 - c.** Snowball
 - d.** S3
- 10.** What service would you use to monitor the specific AWS API calls that are being made against your AWS application?
- a.** CloudInspector
 - b.** CloudFormation
 - c.** CloudTrail
 - d.** CloudWatch

FOUNDATION TOPICS

ADVANTAGES OF CLOUD TECHNOLOGIES

It is no major surprise that various public cloud vendors (led by AWS) are experiencing increased success each and every year. This is not surprising because the list of advantages for the cloud continues to grow! Here are just some:

Key Topic

- ■ **CapEx is replaced by OpEx:** Using public cloud technologies enables start-ups and existing organizations to provide new features and services with a minimum of capital expenditures (CapEx). Instead, public cloud expenses revolve around monthly operating expenses (OpEx). For most organizations, OpEx represents significant advantages when compared to CapEx investments.
- ■ **Lack of contractual commitments:** Many public cloud vendors charge on an hourly basis. Some even offer services and charge per second of usage (AWS has moved several features to this model). For most services, there is no long-term commitment to an organization. You can roll out new projects or initiatives and, if needed, roll back with no contractual commitments long term. This lack of contractual commitment helps increase the agility of IT operations and lowers financial risks associated with innovative technologies.
- ■ **Reduction of required negotiations:** New account establishment with public cloud vendors is simple, and pricing for the major public cloud vendors continually decreases. These factors reduce the need for cost negotiations that were once commonplace with service provider interactions.
- ■ **Reduced procurement delays:** Additional resources can be set up with most cloud implementations within seconds.
- ■ **“Pay as you go” model:** If more resources are needed to support a growing cloud presence, you can get these resources on demand and pay for them only when needed. Conversely, if fewer resources are required, you can run less and pay for only what you need.
- ■ **High levels of security are possible:** Because you can focus on the security of your resources and the cloud provider can focus on their security responsibilities (such as physical security and hypervisor security), the resulting infrastructure can meet stringent levels of security. This security model is appropriately termed the Shared Responsibility model.
- ■ **Flexibility:** Thanks to features in public cloud vendors like AWS, you can quickly scale the cloud-based infrastructure up and down and out and in as needed. This advantage is often termed *elasticity*. Auto scaling functionality inside AWS allows the dynamic creation and destruction of resources based on actual client demand. Such scaling can occur with little to no administrator interaction. By the way, when discussing scaling the resources of a service, we are scaling those resources horizontally (out and in with elasticity), while the service made up of those resources is being scaled up

and down (vertically because the single service is getting bigger or smaller). A single service scales both up and down, and out and in—depending on the context.

- ■ **A massive global infrastructure:** Most of the public cloud vendors now offer resources located all over the globe. This global dispersion of resources serves large multinational organizations well because resources needed for certain parts of the globe can be stored and optimized for access in those regions. Also, companies with clients all over the world can meet with similar access advantages when servicing the needs of clients.
- ■ **SaaS, PaaS, and IaaS offerings:** Cloud technologies have become so advanced that organizations can choose to give applications to clients, development environments, or even entire IT infrastructures using the technologies that make up the cloud. In fact, because the cloud can offer almost any component of IT these days, many refer to the cloud as an Everything as a Service (XaaS) opportunity.
- ■ **Emphasis on API support:** More and more, cloud vendors are taking an application programming interface (API) first approach. This makes the same configuration possible with REST APIs (typically used) that would be possible with an SDK, CLI, or GUI. The API first approach means no interface (CLI or GUI) changes are made until API calls are made first. Thus, there is nothing that cannot be automated!

AN OVERVIEW OF KEY AWS SERVICES

You should leave this chapter well versed in the purpose of some of the vital AWS services. This section breaks down the services for you by category to make them easier to memorize and recommend.

Note

Believe it or not, this section describes just some of the services that AWS has to offer. More are being created by Amazon and being used by customers all the time. Consider making flash cards to help you memorize these services and their role in AWS. After you define the key terms at the end of the chapter (and verify they are correct in the Key Terms Glossary at the end of the book), consider using them as the basis for your flash cards!

Compute Services

Who does not need computer horsepower these days? From the simplest of applications to the most complex (such as artificial intelligence), some level of compute resources is required today. Amazon Web Services features solutions for all different scales and applications as shown by the essential foundational services found in this category.

Elastic Compute Cloud



Amazon Elastic Compute Cloud (EC2) is a web service that gives secure and resizable compute resources in the AWS cloud. The EC2 service allows you to provision and configure capacity with minimal effort. It provides you with easy control of your computing resources.

EC2 reduces the time required to obtain and boot new servers (EC2 instances) to just minutes. This efficiency allows you to scale capacity vertically (up and down—making your server resources bigger or smaller) and horizontally (out and in—adding more capacity in the form of more instances), as your computing requirements change. As described in the previous section, this property is known as elasticity.

Figure 1-1 shows the EC2 dashboard in the AWS management console with two instances currently running.

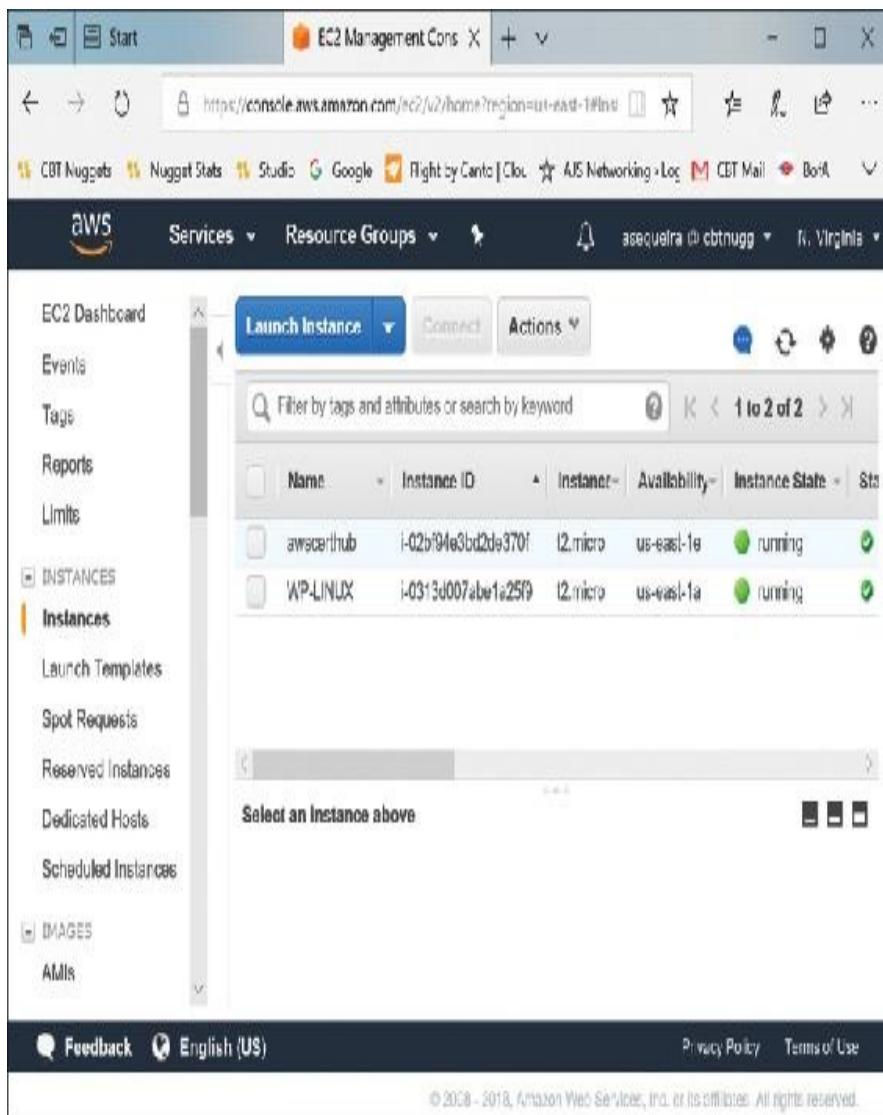


Figure 1-1 The AWS EC2 Dashboard

The many benefits of EC2 in AWS include the following:

- EC2 allows for a controlled expenditure as your business expands; you pay only for the resources you use as you grow.
- EC2 provides you with the tools to build failure-resilient applications that isolate themselves from common failure scenarios.
- EC2 enables you to increase or decrease capacity within minutes, not hours or days. You can commission one, hundreds, or even thousands of server instances simultaneously.
- You have complete control of your EC2 instances. You have root access

to each one, and you can interact with them as you would any traditional virtual machine.

- ■ You can stop your EC2 instance while retaining the data on your boot partition and then subsequently restart the same instance using web service APIs. Instances can be rebooted remotely using web service APIs.
- ■ You can choose among multiple instance types, operating systems, and software packages. Instance types inside AWS permit the choice of emphasis on CPU, RAM, and/or networking resources.
- ■ EC2 integrates with most AWS services, such as Simple Storage Service (S3), Relational Database Service (RDS), and Virtual Private Cloud (VPC). This tight integration allows you to use EC2 for a wide variety of compute scenarios.
- ■ EC2 offers a reliable environment where replacement instances can be rapidly and predictably commissioned. The service runs within Amazon's proven network infrastructure and data centers. AWS offers as much as 99.95 percent availability for each region.
- ■ Amazon EC2 works in conjunction with Amazon VPC to provide security and robust networking functionality for your compute resources:
 - ■ Your compute instances are located in a VPC with an IP address range that you specify.
 - ■ You decide which instances are exposed to the Internet and which remain private.
 - ■ Security groups and network access control lists (ACLs) allow you to control inbound and outbound network access to and from your instances.
 - ■ You can connect your existing IT infrastructure to resources in your VPC using industry-standard encrypted IPsec virtual private network (VPN) connections, or you can take advantage of a private AWS Direct Connect option.
- ■ You can provision your Amazon EC2 resources as dedicated instances. Dedicated instances are Amazon EC2 instances that run on hardware dedicated to a single customer for additional isolation. Alternatively, you can provision your Amazon EC2 resources on dedicated hosts, which are physical servers with EC2 instance capacity entirely dedicated to your use. Dedicated hosts can help you address compliance requirements and reduce

costs by allowing you to use your existing server-bound software licenses.

- ■ Several pricing models exist, including the following:
 - ■ **On-demand instances:** With this model, you pay for compute capacity by the hour (or even by the second with some AMIs) with no long-term commitments. You can increase or decrease your compute capacity depending on the demands of your application and pay the specified hourly rate only for the instances you use. The use of on-demand instances frees you from the costs and complexities of planning, purchasing, and maintaining hardware. As mentioned in the first section, this model also transforms what are commonly substantial fixed costs into much smaller variable costs.
 - ■ **Reserved instances:** This model provides you with a significant discount (up to 75 percent) compared to on-demand instance pricing. You have the flexibility to change families, operating system types, and tenancies while benefitting from reserved instance pricing when you use convertible reserved instances.
 - ■ **Spot instances:** These instances allow you to bid on spare EC2 computing capacity. Because spot instances are often available at a discount compared to on-demand pricing, you can significantly reduce the cost (up to 90 percent) of running your applications.

Lambda



AWS Lambda lets you run code without the burden of provisioning or managing servers. The code you run against Lambda can be for various aspects of an application or service.

When you use Lambda, you upload your code, and Lambda does everything required to run and scale your code with high availability and fault tolerance. Again, you are not required to provision or configure any server infrastructure yourself.

How does your code know when to execute? It does so in several ways:

- ■ Code can automatically trigger from other AWS services.
- ■ You can call it directly from any app.
- ■ You can call it directly from any mobile app.
- ■ You can schedule it with a cron job.
- ■ You can use a custom REST API interface.

Remember, with Lambda, you pay only for the compute time you consume; there is no charge when your code is not running.

Figure 1-2 shows the Lambda service in the AWS management console.

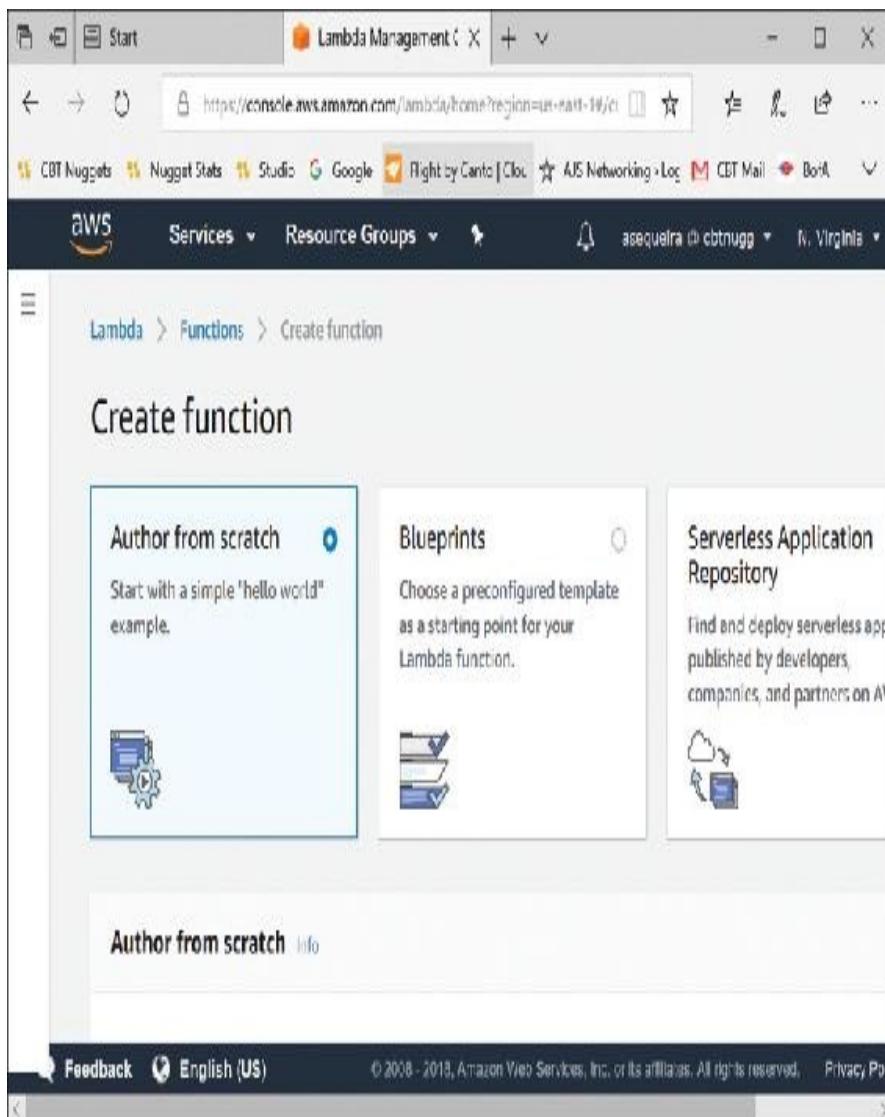


Figure 1-2 AWS Lambda

Elastic Container Service

The Amazon Elastic Container Service (ECS) is a highly scalable, high-performance container management service that supports Docker containers. ECS offers the following features:

- ECS permits you to run applications efficiently on a managed cluster of EC2 instances. It eliminates the need for you to install, operate, and scale your own cluster management infrastructure.
- With simple API calls, you can launch and stop Docker-enabled applications, query the complete state of your cluster, and access many

familiar features:

- ■ Security groups
- ■ Elastic Load Balancing
- ■ Elastic Block Store (EBS) volumes
- ■ Identity and Access Management (IAM) roles
- ■ You can use ECS to schedule the placement of containers across your cluster based on your resource needs and availability requirements.
- ■ You can integrate your own scheduler or third-party schedulers to meet business or application-specific requirements.

Elastic Container Registry

Amazon Elastic Container Registry (ECR) is a fully managed Docker container registry that makes it easy for developers to store, manage, and deploy Docker container images. The ECR offers the following features:

- ■ Integration with the Amazon Elastic Container Service (ECS) simplifies your development to production workflow.
- ■ It eliminates concerns about scaling the underlying infrastructure.
- ■ ECR hosts your images in a highly available and scalable architecture, allowing you to deploy containers for your applications reliably.
- ■ Integration with IAM provides resource-level control of each repository.
- ■ With ECR, you have no upfront fees or commitments. You pay only for the amount of data you store in your repositories and data transferred to the Internet.

Elastic Container Service for Kubernetes

At press time, Amazon Elastic Container Service for Kubernetes (EKS) is in preview only with AWS, but it will certainly be popular. This new, fully managed service is easy for you to use with AWS without having to be an expert in managing Kubernetes clusters. Features include the following:

- ■ EKS runs the upstream version of the open-source Kubernetes software,

so you can use all the existing plug-ins and tools from the Kubernetes community.

- EKS automatically runs K8s with three masters across three AZs to protect against a single point of failure.
- EKS also automatically detects and replaces unhealthy masters, and it provides automated version upgrades and patching for the masters.
- EKS is integrated with a number of key AWS features such as Elastic Load Balancing for load distribution, IAM for authentication, VPC for isolation, PrivateLink for private network access, and CloudTrail for logging.

Fargate

You can consider Fargate the “Lambda” of Docker containers. Fargate is a technology for Amazon ECS and EKS that allows you to run containers without having to manage servers or clusters. This capability removes the need to choose server types, decide when to scale your clusters, or optimize cluster packing.

Serverless Application Repository

The AWS Serverless Application Repository enables you to quickly deploy code samples, components, and complete applications for common use cases such as web and mobile back ends, event and data processing, logging, monitoring, IoT, and more. Each application is packaged with an AWS Serverless Application Model (SAM) template that defines the AWS resources used. Publicly shared applications also include a link to the application’s source code. You can also use the Serverless Application Repository to publish your own applications and share them within your team, across your organization, or with the community at large.

Lightsail

Lightsail provides a simple interface to launch and manage a

virtual private server with AWS. Lightsail configures AWS for your needs and provides a virtual machine, SSD-based storage, data transfer, DNS management, and a static IP address. The price is low and predictable for these services. Figure 1-3 shows Lightsail in AWS.

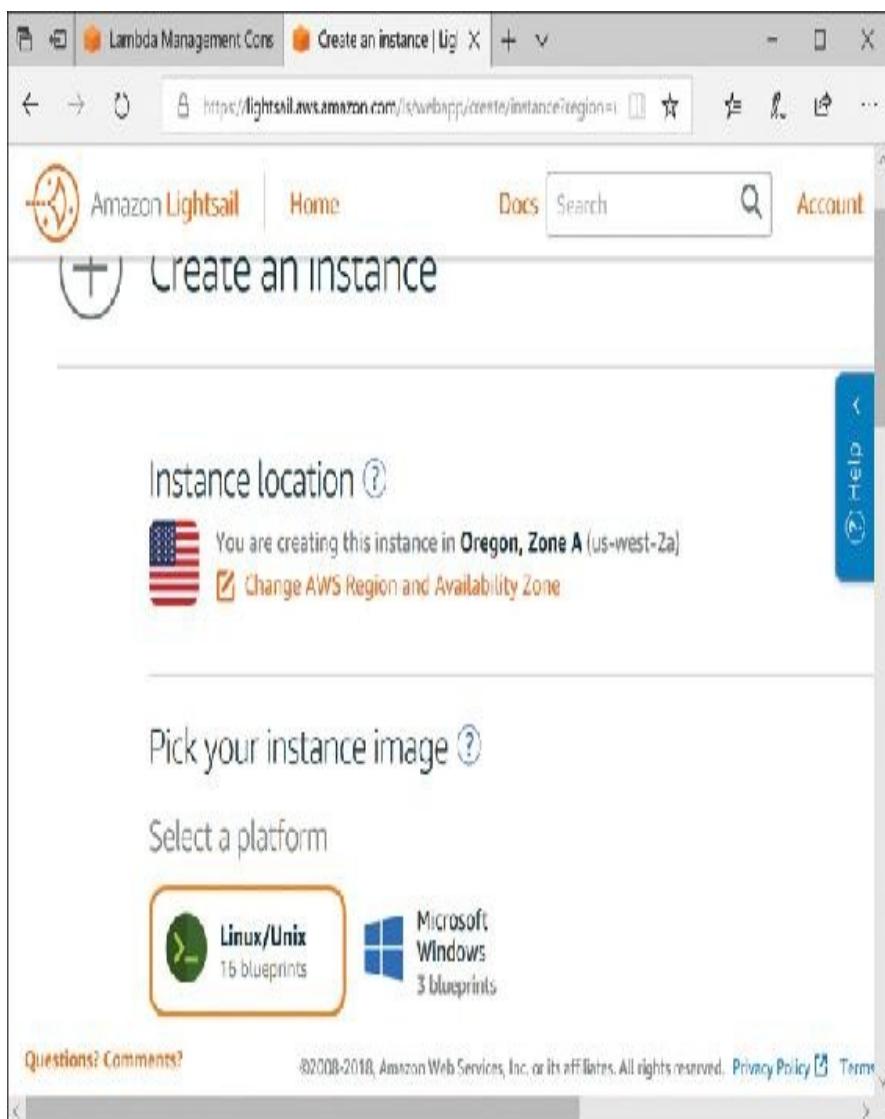


Figure 1-3 AWS Lightsail

AWS Batch

AWS Batch enables developers, scientists, and engineers to

quickly and efficiently run hundreds of thousands of batch computing jobs on AWS. Batch offers the following features:

- ■ Batch dynamically provisions the optimal quantity and compute resources based on the volume and specific resource requirements of the batch jobs submitted.
- ■ It eliminates the need to install and manage batch computing software or server clusters that you use to run your jobs.
- ■ Batch plans, schedules, and executes your batch computing workloads across the full range of AWS compute services and features, such as EC2 and Spot Instances.

Elastic Beanstalk

AWS Elastic Beanstalk is an easy-to-use service for deploying and scaling web applications and services developed with:

- ■ Java
- ■ .NET
- ■ PHP
- ■ Node.js
- ■ Python
- ■ Ruby
- ■ Go
- ■ Docker

These web applications are run on familiar servers such as:

- ■ Apache
- ■ Nginx
- ■ Passenger
- ■ Internet Information Services (IIS)

Amazingly, with this service, you upload your code and Elastic Beanstalk automatically handles the deployment,

including capacity provisioning, load balancing, auto-scaling, and application health monitoring.

Figure 1-4 shows Elastic Beanstalk in AWS.

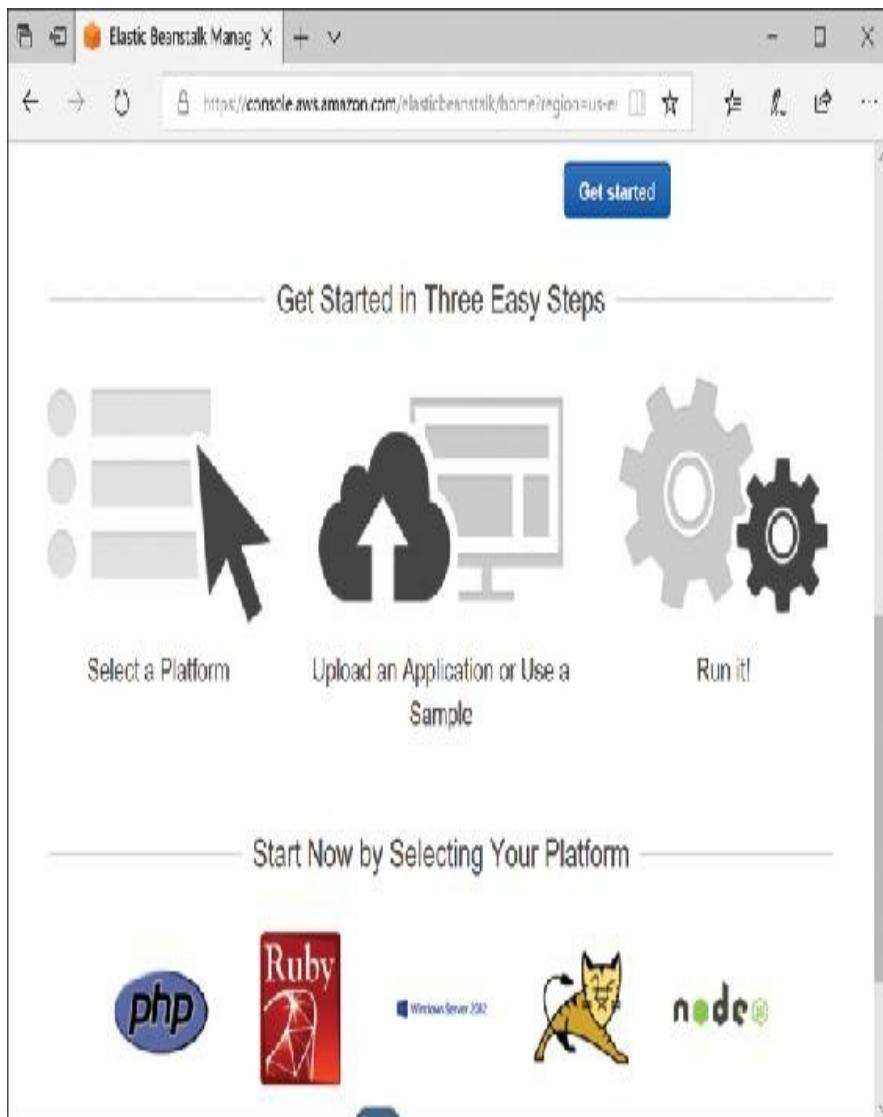


Figure 1-4 Elastic Beanstalk in AWS

Elastic Load Balancing



Elastic Load Balancing (ELB) automatically distributes

incoming application traffic across multiple EC2 instances. ELB enables you to achieve higher levels of fault tolerance in your applications.

Elastic Load Balancing offers the following three types of load balancers that feature high availability, automatic scaling, and robust security:

- **Classic Load Balancer:** Routes traffic based on either application or network-level information; AWS refers to this load balancer service as simply the Legacy Load Balancer now.
- **Application Load Balancer:** Routes traffic based on advanced application-level information that includes the content of the request.
- **Network Load Balancer:** Operates at the Layer 3 (Network) layer of the OSI model as well as Layer 4 (Transport). Instead of using DNS endpoints, it uses IP addresses.

The Legacy (Classic) Load Balancer is ideal for simple load balancing of traffic across multiple EC2 instances, whereas the Application Load Balancer is ideal for applications needing advanced routing capabilities, microservices, and container-based architectures.

The Application Load Balancer also can route traffic to multiple services or load balance across multiple ports on the same EC2 instance. The Application Load Balancer can also work with different *target groups*. They can commonly be used to develop microservices or test builds.

Auto Scaling



Auto Scaling is an excellent deployment option with EC2 instances (among other AWS service choices). It permits you to:

- ■ Maintain application availability.
- ■ Scale your Amazon EC2 capacity automatically according to conditions that you define.
- ■ Set the exact number of EC2 instances you want running at any given time.

Figure 1-5 shows Auto Scaling in the management console.



Figure 1-5 Auto Scaling in AWS

CloudFormation

AWS CloudFormation gives you an easy way to provision and

configure related AWS resources based on a template.

Features include

- ■ Sample templates to describe your AWS resources
- ■ Templates that describe any associated dependencies or runtime parameters required by your application
- ■ Assistance with the order for provisioning AWS services or the subtleties of making those dependencies work in some deployment cases

It also allows you to do the following:

- ■ You can create your own templates.
- ■ After CloudFormation deploys the AWS resources, you can modify and update them in a controlled and predictable way.
- ■ You can also visualize your templates as diagrams and edit them using a drag-and-drop interface with the AWS CloudFormation Designer.

Application Services

Applications make the world go round these days, it seems.

They make up a key element in cloud technologies.

Specifically, applications lie at the heart of Software as a Service (SaaS). This section provides an overview of AWS services targeted at applications.

OpsWorks

AWS OpsWorks is a configuration management service that uses Chef or Puppet. These automation platforms treat server configurations as code. OpsWorks uses Chef or Puppet to automate how servers are configured, deployed, and managed across your EC2 instances or on-premises compute environments.

CloudFront

Amazon CloudFront is a global content delivery network (CDN) service. This service:

- ■ Accelerates delivery of your websites, APIs, video content, or other web assets.
- ■ Automatically routes requests for your content to the nearest edge location, so it delivers content with the best possible performance.
- ■ Is optimized to work with other services in AWS, such as:
 - ■ S3
 - ■ EC2
 - ■ Elastic Load Balancing
 - ■ Route 53
 - ■ Non-AWS origin server that stores the original definitive versions of your files
- ■ Lowers the cost of using S3. Data transfer out costs and operations costs (PUTs and GETs) are cheaper than S3 alone; this will speed up S3 performance by adding a caching layer and reduce S3 costs.
- ■ Lowers the cost of EC2 and other resource uses—again by lowering data transfer out fees but also reducing the load on the back end.

Simple Queue Service

Simple Queue Service (SQS) is a fast, reliable, scalable, distributed, and fully managed message queuing service. It offers the following:

- ■ It decouples the components of a cloud application making it cost-effective and straightforward.
- ■ It transmits any volume of data, without losing messages or requiring other services to be always available.
- ■ It includes standard queues with high throughput and at-least-once processing and first in, first out (FIFO) queues that provide FIFO delivery and exactly once processing.

Simple Notification Service

Simple Notification Service (SNS) is a fast, flexible, distributed, and fully managed push notification service that lets you send individual messages or fan out messages to many

recipients. It enables you to do the following:

- ■ You can send push notifications to mobile device users, email recipients, or even send messages to other distributed services.
- ■ You can send notifications to Apple, Google, Fire OS, and Windows devices, as well as to Android devices in China with Baidu Cloud Push.
- ■ You can use Amazon SNS to send SMS messages to mobile device users worldwide.
- ■ SNS can also deliver messages to Simple Queue Service (SQS), Lambda functions, or to any HTTP endpoint.

Kinesis

Amazon Kinesis makes it easy to collect, process, and analyze real-time streaming data. Amazon Kinesis allows you to process streaming data at any scale and provides the flexibility to choose the tools that best suit the requirements of your application. With Amazon Kinesis, you can ingest real-time data such as video, audio, application logs, website clickstreams, and IoT telemetry data for machine learning, analytics, and other applications. Amazon Kinesis enables you to process and analyze data as it arrives and respond instantly.

Database Services

Many types of databases are available today. The great news is AWS supports these varieties. In fact, AWS permits several different approaches to their implementation. This section gives you an overview of these exciting technologies.

Aurora



Amazon Aurora is a MySQL and PostgreSQL-compatible relational database engine. It offers many benefits, which

include:

- **■ High Performance:** Aurora can provide up to five times the throughput of standard MySQL or twice the throughput of standard PostgreSQL running on the same hardware.
- **■ Highly Secure:** Aurora provides multiple levels of security for your database. These include network isolation using a VPC, encryption at rest using keys you create and control through Key Management Service (KMS), and encryption of data in transit using SSL.
- **■ MySQL and PostgreSQL Compatible:** The Aurora database engine is fully compatible with MySQL 5.6 using the InnoDB storage engine.
- **■ Highly Scalable:** You can scale your Aurora database from an instance with 2 vCPUs and 4 GBs of memory up to an instance with 32 vCPUs and 244 GBs of memory.
- **■ High Availability and Durability:** Aurora is designed to offer higher than 99.99 percent availability.
- **■ Fully Managed:** Aurora is a fully managed database service. Amazon handles tasks such as hardware provisioning, software patching, setup, configuration, monitoring, and backups.

Relational Database Service



Relational Database Service (RDS) makes it easy to set up, operate, and scale a relational database in the cloud. RDS provides six database engines to choose from, including Aurora, PostgreSQL, MySQL, MariaDB, Oracle, and Microsoft SQL Server.

Benefits of RDS include:

- **■ Fast and Easy to Administer:** You can use the AWS Management Console, the AWS RDS command-line interface, or simple API calls to access the capabilities of a production-ready relational database in minutes.
- **■ Highly Scalable:** You can scale your database's compute and storage resources with only a few mouse clicks or an API call, often with no

downtime.

- ■ **Available and Durable:** RDS runs on the same highly reliable infrastructure used by other Amazon Web Services. When you provision a Multi-AZ DB instance, RDS synchronously replicates the data to a standby instance in a different Availability Zone (AZ).
- ■ **Secure:** RDS makes it easy to control network access to your database. RDS also lets you run your database instances in a VPC, which enables you to isolate your database instances and connect to your existing IT infrastructure through an industry-standard encrypted IPsec VPN. Many RDS engine types offer encryption at rest and encryption in transit. You can also take advantage of Direct Connect.
- ■ **Inexpensive:** You pay low rates and only for the resources you consume.

DynamoDB

Amazon DynamoDB is a fast and flexible NoSQL database service for all applications that need consistent, single-digit millisecond latency at any scale. It is a great fit for mobile, web, gaming, ad-tech, Internet of Things (IoT), and many other applications.

Benefits of DynamoDB include:

- ■ **Fast, Consistent Performance:** DynamoDB delivers consistent, fast performance at any scale for all applications.
- ■ **Highly Scalable:** When you create a table, you specify how much request capacity you require. If your throughput requirements change, you update your table's request capacity using the AWS Management Console or the DynamoDB APIs. DynamoDB manages all the scaling behind the scenes, and you are still able to achieve your previous throughput levels while scaling is underway.
- ■ **Fully Managed:** DynamoDB is a fully managed cloud NoSQL database service. You create a database table, set your throughput, and let the service handle the rest.
- ■ **Event-Driven Programming:** DynamoDB integrates with Lambda to provide triggers that enable you to architect applications that automatically react to data changes.

- **Fine-grained Access Control:** DynamoDB integrates with IAM for fine-grained access control.
- **Flexible:** DynamoDB supports both document and key-value data structures, giving you the flexibility to design the best data architecture that is optimal for your application.

ElastiCache

ElastiCache is a web service that makes it easy to deploy, operate, and scale an in-memory cache in the cloud. The service improves the performance of web applications by allowing you to retrieve information from fast, managed, in-memory caches, instead of relying entirely on slower disk-based databases.

ElastiCache supports two open-source in-memory caching engines:

- **Redis:** A fast, open-source, in-memory data store, and cache. ElastiCache for Redis is a Redis-compatible in-memory service that delivers the ease of use and power of Redis along with the availability, reliability, and performance suitable for the most demanding applications.
- **Memcached:** A widely adopted memory object caching system. ElastiCache is protocol-compliant with Memcached, so tools that you use today with existing Memcached environments work seamlessly with the service.

Redshift

Redshift is a fast, fully managed, petabyte-scale data warehouse that makes it simple and cost-effective to analyze all your data using your existing business intelligence tools.

Features include:

- **High Performance:** Redshift obtains a very high query performance on data sets ranging in size from a hundred gigabytes to a petabyte or more.
- **Reduced I/O:** It uses columnar storage, data compression, and zone maps to reduce the amount of I/O needed to perform queries.

- **MPP:** Redshift has massively parallel processing (MPP) data warehouse architecture, parallelizing and distributing SQL operations to take advantage of all available resources. The underlying hardware is designed for high-performance data processing, using locally attached storage to maximize throughput between the CPUs and drives, and a 10GigE mesh network to maximize throughput between nodes.

Database Migration Service

AWS Database Migration Service helps you migrate databases to AWS easily and securely. Features include:

- **Minimize Downtime:** The source database remains fully operational during the migration, minimizing downtime to applications that rely on the database.
- **Broad Database Support:** Database Migration Service migrates your data to and from most widely used commercial and open-source databases. The service supports homogeneous migrations such as Oracle to Oracle, as well as various migrations between different database platforms, such as Oracle to Amazon Aurora or Microsoft SQL Server to MySQL.
- **Streams Data:** Database Migration Service also allows you to stream data to Redshift from any of the supported sources including Aurora, PostgreSQL, MySQL, MariaDB, Oracle, SAP ASE, and SQL Server, enabling consolidation and straightforward analysis of data in the petabyte-scale data warehouse.
- **Continuous Data Replication:** You can use AWS Database Migration Service for continuous data replication with high availability.

Networking Services

The network is the foundation for traditional information technology (IT) infrastructures. AWS provides key network elements as part of their global infrastructure in the form of Virtual Private Clouds (VPCs). While the network is critical, it no longer needs to dominate the attention of the IT department. AWS engages in network abstraction so that you can focus on even more valuable components to your business goals. This section provides a nice overview of key AWS

networking services.

The AWS Global Infrastructure



AWS serves over a million active customers in more than 190 countries. Amazon is steadily expanding global infrastructure to help customers achieve lower latency and higher throughput and to ensure that their data resides only in the region they specify.

Amazon builds the AWS Cloud infrastructure around regions and Availability Zones (AZs):

- A region is a physical location in the world where we have multiple AZs. Note that a region, by design, must have at least two or more AZs, never just one.
- AZs consist of one or more discrete data centers, each with redundant power, networking, and connectivity, housed in separate facilities.
- These AZs enable you to operate production applications and databases that are more highly available, fault-tolerant, and scalable than would be possible from a single data center.
- At the time of this writing, there are 18 regions around the world and 55 AZs. Note that these numbers are now increasing at a faster rate than ever.

Each Amazon region is designed to be completely isolated from the other Amazon regions. This isolation achieves the highest possible fault tolerance and stability. Each AZ is isolated, but Amazon connects the AZs in a region through low-latency links.

AWS provides you with the flexibility to place instances and store data within multiple geographic regions as well as across multiple Availability Zones within each region. Amazon designs each Availability Zone as an independent failure zone.

This independence means that Amazon physically separates Availability Zones within a typical metropolitan region. Amazon chooses lower-risk areas (such as low-risk floodplains) in each region.

In addition to discrete uninterruptible power supply (UPS) and onsite backup generation facilities, they are each fed via different grids from independent utilities to reduce single points of failure further. AZs are all redundantly connected to multiple tier-1 transit providers. Some AZs have their own power substations.

Virtual Private Cloud



Amazon Virtual Private Cloud (Amazon VPC) lets you provision a logically isolated section of the AWS Cloud where you can launch AWS resources in a virtual network that you define. You have complete control over your virtual networking environment, including a selection of your IP address range, the creation of subnets, and the configuration of route tables and network gateways. You can use both IPv4 and IPv6 in your VPC for secure and easy access to resources and applications.

You can create a hardware virtual private network (VPN) connection between your corporate data center and your VPC and leverage the AWS Cloud as an extension of your corporate data center. Note that, by default, a VPC is a completely isolated network, with no ability for traffic to flow in or out of the VPC. For traffic to flow, gateways along with associated route tables, security groups, and NACLs would

need to be provisioned.

Figure 1-6 shows VPC components in the AWS management console.

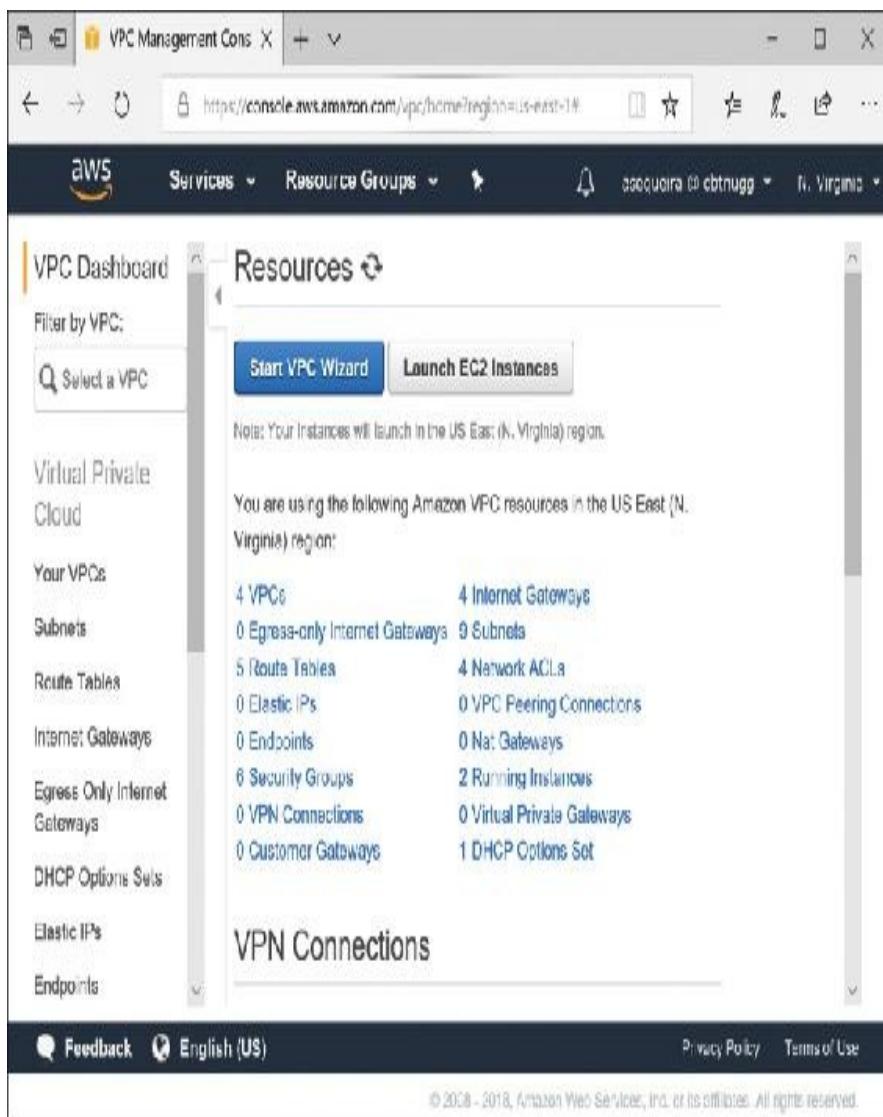


Figure 1-6 VPC Components in AWS

Direct Connect

AWS Direct Connect makes it easy to establish a dedicated network connection from your premises to AWS. Features are numerous and include:

- ■ Establishment of private connectivity between AWS and your data center, office, or colocation environment
- ■ Potential reduction of your network costs
- ■ Potential increase in bandwidth throughput
- ■ Typically, a more consistent network experience than Internet-based connections
- ■ Use of 802.1Q VLANs that enable you to partition the connection into multiple virtual interfaces able to access different resources

Route 53

Amazon Route 53 is a highly available and scalable cloud Domain Name System (DNS) web service.

Amazon Route 53 effectively directs user requests to infrastructure running in AWS—such as EC2 instances, Elastic Load Balancing load balancers, or S3 buckets—and can also be used to route users to infrastructure outside of AWS. You can use Route 53 to configure DNS health checks to route traffic to healthy endpoints or to monitor the health of your application and its endpoints independently.

Route 53 traffic flow makes it easy for you to manage traffic globally through a variety of routing types, including latency-based routing, Geo DNS, and weighted round robin—you can combine all of these with DNS Failover to enable a variety of low-latency, fault-tolerant architectures.

Using Route 53 traffic flow's visual editor, you can efficiently manage how your end users are routed to your application's endpoints—whether in a single AWS region or distributed around the globe. Route 53 also offers Domain Name Registration. You can purchase and manage domain names such as ajsnetworking.com, and Route 53 automatically

configures the DNS settings for your domains.

Storage Services

More and more information is digital these days! This means that you are often responsible for storing more and more data. This section details some key AWS data storage services you might take advantage of.

Simple Storage Service



Amazon Simple Storage Service (Amazon S3) is object storage with a simple web service interface to store and retrieve any amount of data from anywhere on the web. It is designed to deliver 99.99999999 percent durability.

You can use Amazon S3 for a vast number of purposes, such as:

- Primary storage for cloud-native applications
- A bulk repository, or “data lake,” for analytics
- A target for backup and recovery and disaster recovery
- For use with serverless computing

You can move large volumes of data into or out of Amazon S3 with Amazon’s cloud data migration options. You can store data in S3 and then automatically tier the data into lower-cost, longer-term cloud storage classes like S3 Standard–Infrequent Access and Glacier for archiving. You could even utilize a new storage class that reduces high availability (One Zone Infrequent Access) when you do not require it and want to save on storage costs.

There are many advantages to consider thanks to S3. They

include:

- ■ **Simple:** S3 is easy to use with a web-based management console and mobile app. Amazon S3 also provides full REST APIs and SDKs for easy integration with third-party technologies. A command-line interface (CLI) is also extremely popular for working with S3.
- ■ **Durable:** S3 provides a durable infrastructure to store essential data. Amazon designed S3 for durability of 99.99999999 percent of objects. S3 redundantly stores your data across multiple facilities and multiple devices in each facility.
- ■ **Scalable:** With S3, you can store as much data as you want and access it when needed. While there is a 5 TB limit on the size of an individual object, there is no limit to the number of objects you can store!
- ■ **Secure:** S3 supports data transfer over SSL and automatic encryption of your data following the upload. If you want, you can control client-or server-side encryption. You can use Amazon-generated or customer-generated keys and have full key management capabilities/options. You can also configure bucket policies to manage object permissions and control access to your data using IAM.
- ■ **Available:** S3 Standard is designed for up to 99.99 percent availability of objects over a given year and is backed by the Amazon S3 service-level agreement, ensuring that you can rely on it when needed. You can also choose an AWS region to optimize for latency, minimize costs, or address regulatory requirements.
- ■ **Low Cost:** S3 allows you to store large amounts of data at a small cost. Using lifecycle policies, you can configure the automatic migration of your data to different storage tiers within AWS.
- ■ **Simple Data Transfer:** Amazon provides multiple options for cloud data migration and makes it simple and cost-effective for you to move large volumes of data into or out of S3. You can choose from network-optimized, physical disk-based, or third-party connector methods for import to or export from S3.
- ■ **Integrated:** S3 is deeply integrated with other AWS services to make it easier to build solutions that use a range of AWS services. Integrations include:
 - ■ CloudFront

- ■ CloudWatch
- ■ Kinesis
- ■ RDS
- ■ Glacier
- ■ EBS
- ■ DynamoDB
- ■ Redshift
- ■ Route 53
- ■ EMR
- ■ VPC
- ■ Key Management Service (KMS)
- ■ Lambda
- ■ **Easy to Manage:** S3 Storage Management features allow you to take a data-driven approach to storage optimization, data security, and management efficiency. These enterprise-class capabilities give you data about your data so that you can manage your storage based on that personalized metadata.

Elastic Block Store



Amazon Elastic Block Store (EBS) provides persistent block storage volumes for use with EC2 instances in the AWS Cloud. Each Amazon EBS volume is automatically replicated within its Availability Zone to protect you from component failure, offering high availability and durability.

EBS volumes offer the consistent and low-latency performance needed to run your workloads. With Amazon EBS, you can scale your usage up or down within minutes—all while paying a low price for only what you provision.

Features of EBS include:

- **High-Performance Volumes:** Choose between solid-state disk (SSD)-backed or hard disk drive (HDD)-backed volumes that can deliver the performance you need for your most demanding applications.
- **Availability:** Each Amazon EBS volume is designed for 99.999 percent availability and automatically replicates within its Availability Zone to protect your applications from component failure.
- **Encryption:** Amazon EBS encryption provides seamless support for data-at-rest and data-in-transit between EC2 instances and EBS volumes.
- **Access Management:** Amazon's flexible access control policies allow you to specify who can access which EBS volumes, ensuring secure access to your data.
- **Snapshots:** You can protect your data by creating point-in-time snapshots of EBS volumes, which are backed up to Amazon S3 for long-term durability.

Elastic File System

Amazon Elastic File System (EFS) provides simple, scalable file storage for use with Amazon EC2 instances in the AWS Cloud. Amazon EFS is easy to use and offers a simple interface that allows you to create and configure file systems quickly and easily.

Features include:

- **Elastic Storage:** Automatically growing and shrinking as you add and remove files. This elasticity allows your applications to have the storage they need when they need it.
- **Standards-based:** Standard file system interface and file system access semantics when mounted on EC2 instances.

Note

EFS uses NFS v4 and v4.1 standards. Because of the use of NFS v4/4.1, Windows currently does not work with EFS because Windows does not support these protocols. As a result, you use Linux instances with EFS.

- ■ **Supports On-Prem Deployment:** You can mount your Amazon EFS file systems on your on-premises data center servers when connected to your VPC with AWS Direct Connect.
- ■ **Supports Hybrid Cloud:** You can mount your Amazon EFS file systems on on-premises servers to migrate data sets to EFS, enable cloud-bursting scenarios, or back up your on-premises data to EFS.
- ■ **HA and Durability:** EFS is designed for high availability and durability and provides performance for a broad spectrum of workloads and applications, including big data and analytics, media processing workflows, content management, web serving, and home directories.

Glacier

Amazon Glacier is a secure, durable, and extremely low-cost storage service for data archiving and long-term backup. With Glacier, you can:

- ■ Reliably store large or small amounts of data for as little as \$0.004 per gigabyte per month
- ■ Save money compared to on-premises storage options
- ■ Keep costs low yet suitable for varying retrieval needs
- ■ Choose from three options for access to archives, from a few minutes to several hours

Snowball

AWS Snowball is a petabyte-scale data transport solution that uses secure appliances to transfer large amounts of data into and out of AWS.

With Snowball, you do not need to write any code or purchase any hardware to transfer your data. You follow these steps:

Step 1. Create a job in the AWS Management Console.

Step 2. A Snowball appliance is automatically shipped to you.

Step 3. After it arrives, attach the appliance to your local

network, download and run the Snowball client to establish a connection, and then use the client to select the file directories that you want to transfer to the appliance.

Step 4. The client encrypts and transfers the files to the appliance at high speed.

Step 5. Once the transfer is complete and the appliance is ready to be returned, the E Ink shipping label automatically updates. You can track the job status using the Simple Notification Service (SNS), checking text messages, or directly using the console.

Snowball uses multiple layers of security designed to protect your data including tamper-resistant enclosures, 256-bit encryption, and an industry-standard Trusted Platform Module (TPM) designed to ensure both security and the full chain of custody of your data. Once the data transfer job has been processed and verified, AWS performs a software erasure of the Snowball appliance using industry secure erasure standards.

AWS Storage Gateway

The [AWS Storage Gateway](#) service seamlessly enables hybrid storage between on-premises storage environments and the AWS Cloud. Features include:

- **High Performance:** AWS Storage Gateway combines a multiprotocol storage appliance with highly efficient network connectivity to Amazon cloud storage services, delivering local performance with virtually unlimited scale.
- **Hybrid Cloud Support:** You can use it in remote offices and data centers for hybrid cloud workloads involving migration, bursting, and storage

tiering.

Security Services

As enterprises rely on IT and the cloud for more functions than ever, security becomes more important than ever. This is especially true because attackers continuously increase their level of sophistication. This section provides an overview of key AWS security-related services.

Identity and Access Management



AWS Identity and Access Management (IAM) enables you to securely control access to AWS services and resources for your users. Using IAM, you can create and manage AWS users and groups, and use permissions to allow and deny their access to AWS resources. IAM allows you to do the following:

- **Manage IAM users and their access:** You can create users in IAM, assign them individual security credentials (access keys, passwords, and multifactor authentication devices), or request temporary security credentials to provide users access to AWS services and resources. You can manage permissions to control which operations users can perform.
- **Manage IAM roles and their permissions:** You can create roles in IAM and manage permissions to control which operations can be performed by the entity, or AWS service, that assumes the role. You can also define which entity can assume the role.
- **Manage federated users and their permissions:** You can enable identity federation to allow existing identities (users, groups, and roles) in your enterprise to access the AWS Management Console, call AWS APIs, and access resources, without the need to create an IAM user for each identity.

Figure 1-7 shows IAM in the AWS management console.

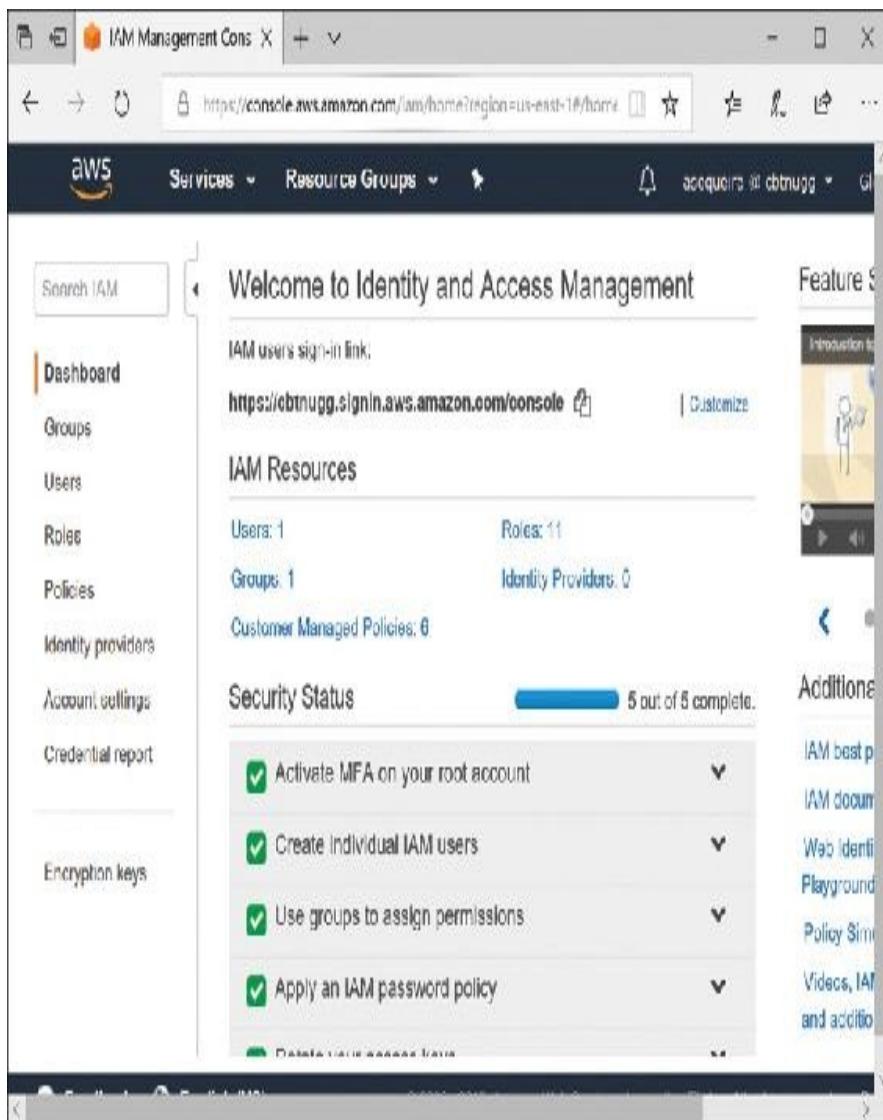


Figure 1-7 IAM in AWS

Web Application Firewall

AWS Web Application Firewall (WAF) helps protect your web applications from common web exploits that could affect application availability, compromise security, or consume excessive resources. Features of AWS WAF include:

- Control over which traffic to allow or block for your web application by defining customizable web security rules
- The creation of custom rules that block common attack patterns, such as SQL injection or cross-site scripting

- ■ The creation of rules you design for your specific application
- ■ The creation of rules with AWS Rule Subscriptions that help prevent against zero-day attacks thanks to dynamic rule creation
- ■ New rules that can be deployed within minutes, letting you respond quickly to changing traffic patterns
- ■ A full-featured API that you can use to automate the creation, deployment, and maintenance of web security rules

Key Management Service

AWS Key Management Service (KMS) is a managed service that makes it easy for you to create and control the encryption keys used to encrypt your data. KMS offers:

- ■ Use of Hardware Security Modules (HSMs) to protect the security of your keys
- ■ Integration with several other AWS services to help you protect the data you store with these services
- ■ Integration with CloudTrail to provide you with logs of all key usage to help meet your regulatory and compliance needs

Directory Services

AWS Directory Service for Microsoft Active Directory (Enterprise Edition), also known as AWS Microsoft AD, enables your directory-aware workloads and AWS resources to use managed Active Directory in the AWS Cloud. There is now also a Standard Edition that allows the SMB space to take advantage of this feature at a more manageable cost. Note the following about Directory Services in AWS:

- ■ The AWS Microsoft AD service is built on actual Microsoft Active Directory and does not require you to synchronize or replicate data from your existing Active Directory to the cloud.
- ■ You can use standard Active Directory administration tools to take advantage of built-in Active Directory features such as Group Policy, trusts, and single sign-on.

- With Microsoft AD, you can quickly join EC2 and RDS for SQL Server instances to a domain and use AWS Enterprise IT applications such as Amazon WorkSpaces with Active Directory users and groups.

Management Services

While most services of AWS allow fine-grained management using a variety of tools, some services are designed for comprehensive management of AWS designs. This section describes these management services at a high level.

Trusted Advisor

AWS Trusted Advisor is an online resource to help you reduce cost, increase performance, and improve security by optimizing your AWS environment.

Trusted Advisor provides real-time guidance to help you provision your resources following AWS best practices.

CloudWatch



Amazon CloudWatch is a monitoring service for AWS Cloud resources and the applications you run on AWS. CloudWatch can:

- Collect and track metrics, collect and monitor log files, set alarms, and automatically react to changes in your AWS resources
- Monitor AWS resources such as EC2 instances, DynamoDB tables, and RDS DB instances
- Monitor custom metrics generated by your applications and services, and any log files your applications generate
- Gain systemwide visibility into resource utilization, application performance, and operational health

CloudTrail

Key Topic

AWS CloudTrail is a web service that records AWS API calls for your account and delivers log files to you. With CloudTrail you have:

- ■ Detailed reports of recorded information, which can include the identity of the API caller, the time of the API call, the source IP address of the API caller, the request parameters, and the response elements returned by the AWS service
- ■ A history of AWS API calls for your account, including API calls made using the AWS Management Console, AWS SDKs, command-line tools, and higher-level AWS services (such as CloudFormation)
- ■ The AWS API call history produced by CloudTrail, which enables security analysis, resource change tracking, and compliance auditing

EXAM PREPARATION TASKS

As mentioned in the section “How to Use This Book” in the Introduction, you have a couple of choices for exam preparation: the exercises here, Chapter 16, “Final Preparation,” and the exam simulation questions in the Pearson Test Prep practice test software.

REVIEW ALL KEY TOPICS

Review the most important topics in this chapter, noted with the Key Topics icon in the margin of the page. Table 1-2 lists a reference of these key topics and the page numbers on which each is found.

Key Topic

Table 1-2 Key Topics for Chapter 1

Key Topic Element	Description	Page Number
List	Advantages of Cloud Technologies	6
Section	Elastic Compute Cloud	8
Section	Lambda	10
Section	Elastic Load Balancing	16
Section	Auto Scaling	16
Section	Aurora	20
Section	Relational Database Service	20
Section	The AWS Global Infrastructure	23

Section	Virtual Private Cloud	24
Section	Simple Storage Service	26
Section	Elastic Block Store	28
Section	Identity and Access Management	31
Section	CloudWatch	34
Section	CloudTrail	34

COMPLETE TABLES AND LISTS FROM MEMORY

There are no memory tables in this chapter.

DEFINE KEY TERMS

Define the following key terms from this chapter and check your answers in the glossary:

Elasticity

CapEx

OpEx

Elastic Compute Cloud (EC2)

Lambda

Elastic Container Service

Elastic Container Registry

Fargate

Elastic Container Service for Kubernetes

Serverless Application Repository

Lightsail

AWS Batch

Elastic Beanstalk

Elastic Load Balancing

Auto Scaling

CloudFormation

OpsWorks

CloudFront

Simple Queue Service (SQS)

Simple Notification Service (SNS)

Kinesis

Aurora

Relational Database Service (RDS)

DynamoDB

ElastiCache

Redshift

Database Migration Service

AWS Global Infrastructure

Virtual Private Cloud (VPC)

Direct Connect

Route 53

Simple Storage Service (S3)

Elastic Block Store

Elastic File System

Glacier

Snowball

AWS Storage Gateway

Identity and Access Management (IAM)

Web Application Firewall (WAF)

Key Management Service (KMS)

Directory Services

Trusted Advisor

CloudWatch

CloudTrail

Q&A

The answers to these questions appear in Appendix A. For more practice with exam format questions, use the Pearson Test Prep practice test software.

1. Name at least five advantages to cloud technologies today.

2. Describe EC2 in AWS.

3. Describe S3 in AWS.

4. Describe a VPC in AWS.

5. Describe the AWS Global Infrastructure.

6. Describe at least four database technologies offered by AWS.

Chapter 2. Designing Resilient Storage

This chapter covers the following subjects:

- **■ Designing Resilient S3 Services:** The chapter begins with the foundational storage service of AWS—S3 object-based storage, which has many resiliency features built right in!
- **■ Designing Resilient EBS Services:** Elastic Block Store (EBS) is a simple way to create block storage for those cloud devices that require it. A classic use case is the EC2 virtual machine disk drives for the operating systems and data required for these systems. This part of the chapter introduces you to EBS and walks you through creating these volumes.
- **■ Designing Resilient EFS Services:** Need a file system compatible with Network File System (NFS) in the cloud? No worries, thanks to the Elastic File System (EFS). This section walks you through using EFS in AWS for resilient file systems.
- **■ Designing Resilient Glacier Services:** You might want a cost-effective backup or archival storage in AWS that is still very resilient. AWS Glacier provides this functionality in the public cloud. This part of the chapter details vital characteristics of Glacier and walks you through the creation and use of a Vault inside this service.

The more enterprises become reliant on technology and IT, the more data there is to store. This chapter covers S3, EBS, EFS, and Glacier with a focus on the resiliency of these storage services. While you learn to set up these storage structures in this chapter, later chapters revisit these topics with a focus on performance and security.

“DO I KNOW THIS ALREADY?” QUIZ

The “Do I Know This Already?” quiz allows you to assess whether you should read the entire chapter. Table 2-1 lists the

major headings in this chapter and the “Do I Know This Already?” quiz questions covering the material in those headings so you can assess your knowledge of these specific areas. The answers to the “Do I Know This Already?” quiz appear in [Appendix A](#).

Table 2-1 “Do I Know This Already?” Foundation Topics
Section-to-Question Mapping

Foundation Topics Section	Questions
Designing Resilient S3 Services	1–4
Designing Resilient EBS Services	5–6
Designing Resilient EFS Services	7–8
Designing Resilient Glacier Services	9–10

Caution

The goal of self-assessment is to gauge your mastery of the topics in this chapter. If you do not know the answer to a question or are only partially sure of the answer, you should mark that question as wrong for purposes of the self-assessment. Giving yourself credit for an answer you correctly guess skews your self-assessment results and might provide you with a false sense of security.

1. What is the durability for S3 storage?

- a.** 99.999%
- b.** 99.99999999%
- c.** 99.9%
- d.** 99.9999%

2. Which of the following is not a class of S3 storage in AWS?

- a.** Glacier
- b.** Standard
- c.** Standard-IA
- d.** Four-Zone-IA

3. What is the inherent approach to durability and availability used by most of the S3 storage classes?

- a.** To automatically store redundant copies of objects across multiple facilities
- b.** To automatically store redundant copies of objects in different Regions
- c.** To automatically store redundant copies of objects on different physical devices in the same physical data center
- d.** To create versions of files by default in all buckets

4. What storage class of S3 would cause you to run the risk of losing data should an AZ of AWS become totally destroyed?

- a.** One-Zone-IA

- b.** Standard-IA
 - c.** Glacier
 - d.** Standard
- 5.** What type of EBS volume would be a good choice for infrequently accessed workloads in an environment where cost savings is a key consideration?
 - a.** Throughput Optimized HDD (st1)
 - b.** Cold HDD (sc1)
 - c.** General Purpose SSD (gp2)
 - d.** Provisioned IOPS SSD (io1)
- 6.** What is the maximum size of an EBS general-purpose SSD volume?
 - a.** 2 TB
 - b.** 8 TB
 - c.** 16 TB
 - d.** 32 TB
- 7.** What protocol should you permit in your security group if you have an EC2 instance that you intend to access EFS?
 - a.** SSH
 - b.** NFS
 - c.** TCP
 - d.** FCoE
- 8.** Which statement about EFS is false?
 - a.** EFS can only be accessed by one EC2 instance at a

time.

- b.** NFSv4 is supported.
- c.** Durability is ensured through the use of multiple AZs.
- d.** EFS is simple to scale.

9. What is the durability of Glacier?

- a.** 99.999%
- b.** 99.99999999%
- c.** 99.9%
- d.** 99.9999%

10. What is the logical container to store an archive inside of Glacier?

- a.** Bucket
- b.** Shelf
- c.** Vault
- d.** Warehouse

FOUNDATION TOPICS

DESIGNING RESILIENT S3 SERVICES

The Simple Storage Service (S3) of AWS is pretty amazing.
Consider these major advantages:



- ■ **Easy access:** Amazon built S3 with ease of access in mind—from Internet to private network writing of data and retrieval.
- ■ **Durability:** A key quality for our focus in this chapter; durability means

the chance of data loss. As you'll learn soon, AWS features amazing durability levels.

- ■ **High availability:** Moving beyond durability for even greater resiliency is the fact that S3 can automatically store copies of your data in different Availability Zones (AZs). In the event of a disaster, another copy is readily available.
- ■ **Scalability:** Your S3 buckets have no limit to their overall size. The limitations become storage costs, which are typically much less than traditional on-premises options. Object sizes can range from 0 bytes to 5 TB; multipart upload is required for single objects greater than 5 GB.
- ■ **Flexibility:** You can redesign and change your storage strategy with ease using S3 buckets and “folders” inside these buckets. Keep in mind the folders are not those from traditional file systems; they are just flexible organizational components to provide storage locations inside the bucket. This chapter elaborates on this issue in greater detail.
- ■ **Pay as you go:** You could easily think of this feature as pay as you “grow” because you can initially be in a free-tier level of storage and then start paying for storage as it increases—of course, paying less as your needs diminish.
- ■ **Multiple classes:** S3 offers four storage classes to help you design resilient storage that is also cost effective. Because this is such critical information for a solutions architect, we cover it in detail next.

S3 Storage Classes

It is critical to learn the different storage options (classes) you can use in S3. The four options are:



- ■ **S3 Standard:** This class offers the following characteristics and features:
 - ■ Durability of 99.99999999 percent across at least three facilities
 - ■ Low latency
 - ■ Resiliency in the event of entire AZ destruction
 - ■ 99.99 percent availability

- ■ An SLA governing performance
- ■ Encryption of in-transit and at-rest data (optional)
- ■ Lifecycle management (optional)
- ■ **S3 Standard-Infrequent Access:** This class offers the following characteristics and features:
 - ■ Many of the S3 standard features, such as incredible durability
 - ■ Designed for less frequent access but still provides responsiveness
 - ■ An ideal class for workloads like backups
 - ■ Designed for 99.9 percent availability
- ■ **S3 One Zone-Infrequent Access:** This class offers the following characteristics and features:
 - ■ This might be an ideal class for objects like secondary backup copies of data.
 - ■ The objects in this storage class could be replicated to another region should the need arise to increase the availability.
 - ■ As you would guess, availability here is reduced slightly and is designed for 99.5 percent.
 - ■ There is a risk of data loss in the event of an AZ's destruction.
- ■ **S3 Glacier:** This class offers the following characteristics and features:
 - ■ This is an ideal storage class for the archiving of data.
 - ■ Access times to archived data can be from minutes to hours depending on your configuration and purchase plan.

Table 2-2 summarizes critical information about S3 and these storage classes with an emphasis on resiliency. In later chapters, we concentrate on high availability and cost optimization, as well as performance when it comes to S3.

Table 2-2 The S3 Classes

S3	S3	S3 One	S3 Glacier

	Standard	Standard-IA	Zone-IA	
Durability	99.9999999 99%	99.9999999 99%	99.9999999 99%	99.9999999 99%
Availability SLA	99.99%	99.9%	99.5%	N/A
Availability Zones	3	3	1	3
Storage Type	Object	Object	Object	Object
Lifecycle Transitions	Yes	Yes	Yes	Yes

Note

What if you are using the S3 Standard class and there are only two Availability Zones in your region? Amazon calls upon a “facility” in this case to help ensure availability. While Amazon does not officially categorize a “facility” as an AZ, we use AZs in the previous table for simplicity of presentation.

Lab: Creating an S3 Bucket

Creating an S3 bucket with the AWS Management Console requires these steps:



Step 1. Sign in to the AWS Management Console and search for S3.

Step 2. Choose **Create Bucket**.

Step 3. On the Name and Region page (see [Figure 2-1](#)), type a name for your bucket and choose the AWS Region where you want the bucket to reside.

- ■ For Bucket Name, type a unique DNS-compliant name for your new bucket. Follow these naming guidelines:
 - ■ The name must be unique across all existing bucket names in Amazon S3; for this lab, consider a name that will almost certainly be unique in AWS, for example, your last name followed by some random digits.
 - ■ The name must not contain uppercase characters.
 - ■ The name must start with a lowercase letter or number.
 - ■ The name must be between 3 and 63 characters long.
 - ■ After you create the bucket, you cannot change the name.

Note

You may want to consider a bucket name that reflects the objects in the bucket. The bucket name is visible in the URL that points to the objects that you are going to put in your bucket. If some obfuscation is more appropriate in high-security environments, you might take an opposite approach and generate randomized names.

- For Region, choose the AWS Region where you want the bucket to reside. Consider a Region close to users of the storage to minimize latency and costs or to address regulatory requirements. Objects stored in a Region never leave that Region unless you explicitly transfer them to another Region.

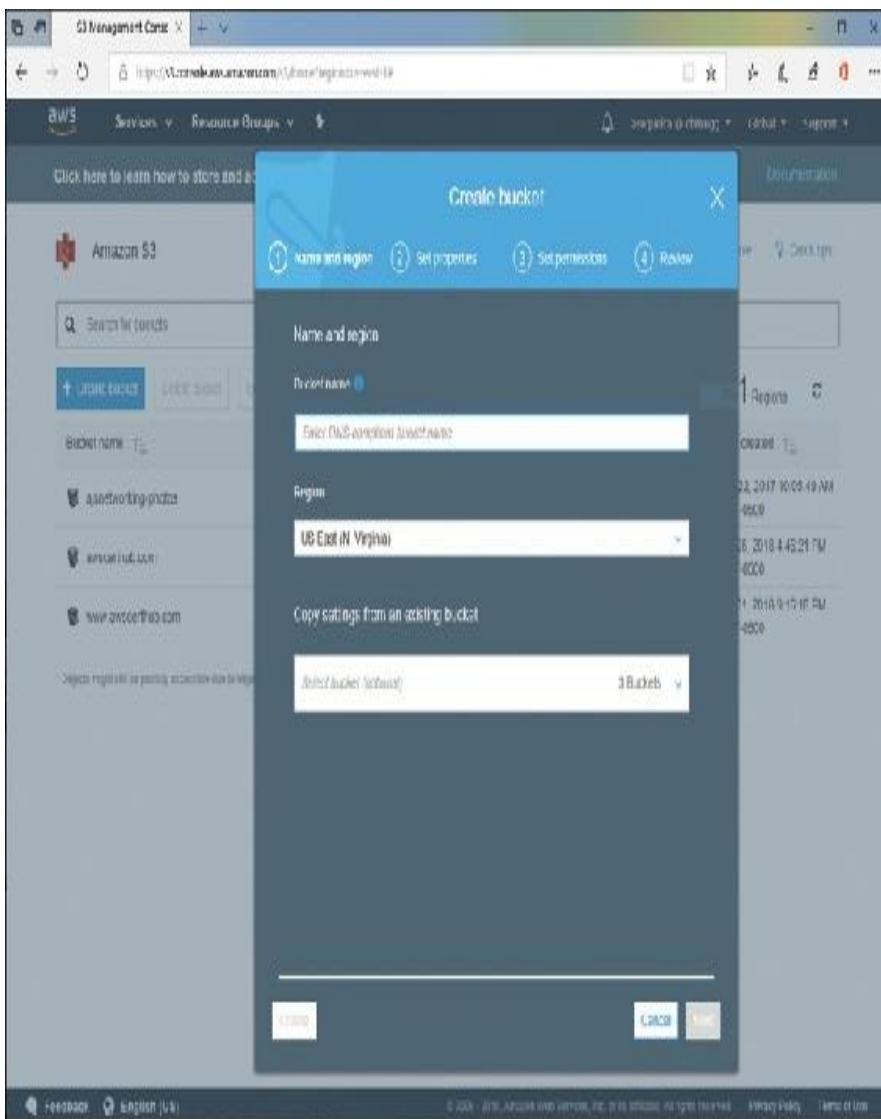


Figure 2-1 Creating an S3 Bucket

Note

AWS is continually tweaking its Management Console interface. Don't be alarmed if your screen appears with slight differences.

Step 4. (Optional) If you have already set up a bucket with the same settings that you want to use for a new bucket that you want to create, you can set it up quickly by choosing **Copy Settings from an Existing Bucket** and then choosing the bucket whose settings you want to copy. The settings for the following bucket properties are copied: versioning, tags, and logging. If you copied settings from another bucket, choose **Create** and you are done with bucket creation.

Step 5. On the Set Properties page, you can configure the following properties for the bucket. Alternatively, you can configure these properties later, after you create the bucket.

- **Versioning:** Versioning enables you to keep multiple versions of an object in one bucket. Versioning is disabled for a new bucket by default; note that this and other settings are disabled by default in an effort to help control costs.
- **Server Access Logging:** Server access logging provides detailed records for the requests that are made to your bucket. By default, Amazon S3 does not collect server access logs.
- **Tags:** With AWS cost allocation, you can use tags to annotate billing for your use of a bucket. A tag is a key-value pair that represents a label that you assign to a bucket. To add tags, choose **Tags** and then choose **Add Tag**.
- **Object-Level Logging:** Object-Level Logging records object-level API activity by using CloudTrail data events.
- **Default Encryption:** Amazon S3 default encryption provides a way to set the default encryption behavior for an S3 bucket; you can set default encryption on a bucket so that all objects are encrypted when they are stored in the bucket.

Step 6. Choose **Next**.

Step 7. On the Set Permissions page, you manage the permissions that are set on the bucket that you are creating. You can grant read access to your bucket to the general public (everyone in the world). Granting public read access is applicable to a small subset of use cases such as when buckets are used for websites. It is not recommended that you change the default setting of Do Not Grant Public Read Access to This Bucket. You can change permissions after you create the bucket. You can also later simply grant individual objects public read access as required without making the whole bucket public. When you are done configuring permissions on the bucket, choose **Next**.

Step 8. On the Review page, verify the settings. If you want to change something, choose **Edit**. If your current settings are correct, choose **Create Bucket**.

Lab Cleanup

Step 1. Navigate to the S3 service in the Management Console.

Step 2. Highlight the bucket you created in this lab. To highlight the bucket, click the whitespace to the right of the hyperlink for the bucket name.

Step 3. Click the **Delete Bucket** button above the list of buckets.

Step 4. Confirm the bucket name for deletion.

DESIGNING RESILIENT EBS SERVICES

In Amazon Web Services, you know you have S3 for object-based storage. But what about block-based storage? AWS provides Elastic Block Store (EBS), which allows you to create block storage for components like EC2 instances to have the needed storage (drives). So if, for example, you want to spin up a Linux instance in EC2, you can store it on Elastic Block Storage, and you will have nice persistent storage for the EC2 instance.

EBS Versus Instance Stores

Technically, there is an alternative to Elastic Block Store when it comes to block storage, and that is called an *instance store*. Sometimes you will see this described in AWS documentation as an *ephemeral store*, meaning that it is temporary and that you do not have persistence with this type of block storage.

When you enter EC2 and select a new Amazon Machine Image (AMI) to spin up an EC2 instance, notice that every single root device type that is used will be EBS. So, Elastic Block Store will be the default for your storage.

You might see this storage type in AWS when you are using the community AMIs. With many of them, you can choose the instance store ephemeral storage type. As you might guess, you get few options when you are using an ephemeral instance store for your EC2 instances. AWS creates the storage when needed (for the EC2 instance) and then destroys it when the EC2 instance is no longer needed. In fact, when using instance stores, you cannot stop and start your EC2 instances. You can only create and terminate (delete) them. What you might find interesting about this instance store approach is that it is how all EC2 instances used to run before the invention of the

Elastic Block Store Service. Keep in mind that ephemeral storage still has its place in AWS because it can be faster and cheaper than EBS. In fact, EBS can get quite expensive over time, and this might present unnecessary costs if you do not require persistence.

Elastic Block Store

Today, the Elastic Block Store is a great place to store the stuff that is required to run your virtual machines. With the EBS approach, you can even make sure that storage sticks around and is persistent when you terminate the VM that's running on top of it.

An EBS volume is often created as part of the creation of a new EC2 instance. When you get to the storage option during EC2 creation, you choose the size and type of EBS storage you require for the EC2 instance.

Fortunately, after creation, Elastic Block Store volumes are simple to work with in Amazon Web Services. For example, say you have a Windows Server running in EC2. When you examine the properties of this instance, you can see the EBS volume. It will have a name like *devsda1*.

However, say you're interested in provisioning a new volume for a specialized purpose on a Windows Server system. This is not a problem thanks to EBS Volumes.

Lab: Creating an EBS Volume

Follow these steps to create a new EBS volume for an EC2 instance:



Step 1. Use the Search feature of the Management Console to search for EC2. Click the link. While in the EC2 Dashboard, click the link for **Elastic Block Store**. In the left-hand column, click **Volumes** (see Figure 2-2).

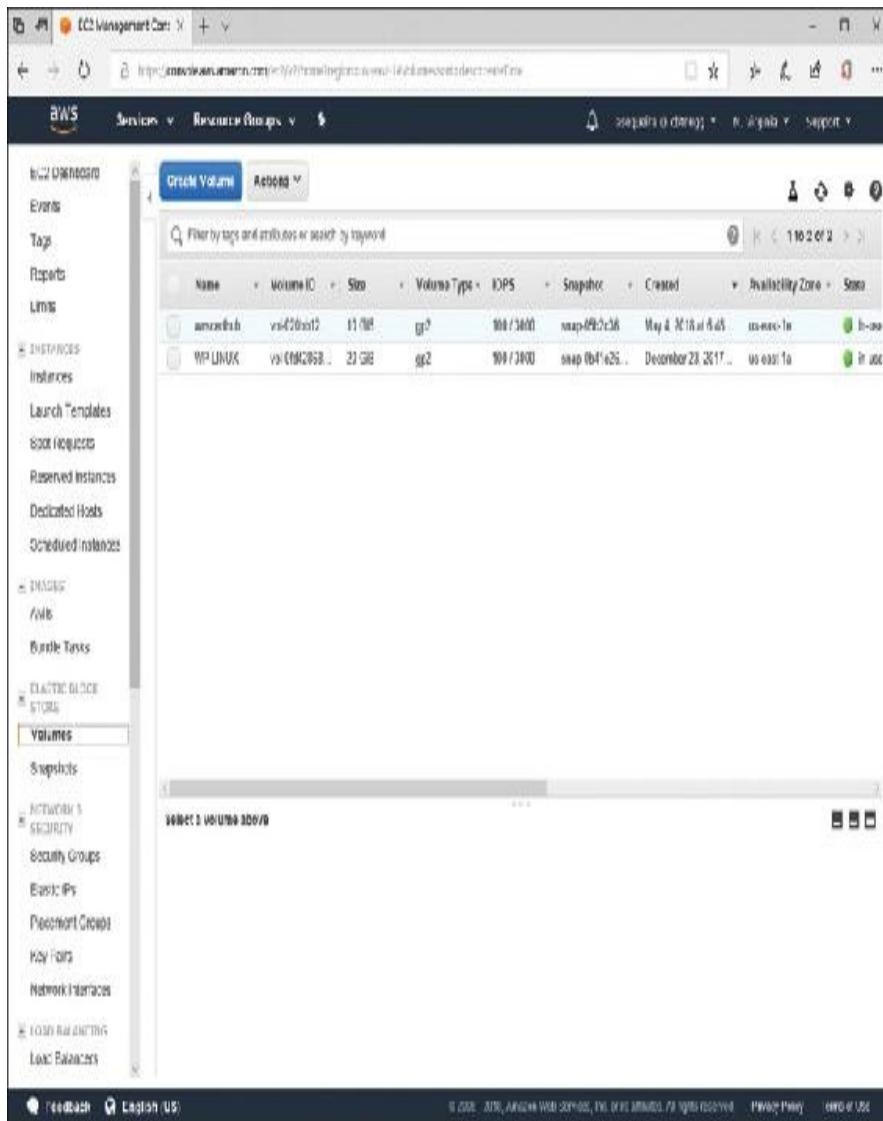


Figure 2-2 Creating a New EBS Volume

Step 2. Click **Create Volume**.

Step 3. Choose the volume type **Cold HDD (SC1)**. Later

in this section, we discuss the different EBS volume options.

Step 4. Choose the size and Availability Zone for the volume.

If you attach the new volume to a running EC2 instance, the storage appears dynamically inside the virtual machine operating system.

Lab Cleanup

Step 1. In the Volumes console, select your volume and choose **Delete Volume** from the Actions drop-down menu.

Step 2. Confirm deletion in the next window.

If you decide you no longer need the attached EBS volume, you can delete it at any time. Go in under the Volume Actions and detach the volume there. Moreover, now you can go up from the Actions menu and quickly delete this volume.

Working with these EBS Volumes is simple inside Amazon Web Services.

Elastic Block Store



All EBS volumes are not created equal. You have the option to create volumes of various types. The choices permit you to balance performance with costs. Table 2-3 lists the current EBS volume types.

Table 2-3 EBS Volume Types

Volume Type	EBS Provisioned IOPS SSD (io1)	EBS General Purpose SSD (gp2)	Throughput Optimized HDD (st1)	Cold HDD (sc1)
Short Description	Highest performance SSD volume designed for latency-sensitive transactional workloads	General Purpose SSD balances price performance for a wide variety of a wide variety of transactional workloads	HDD volume designed for frequently accessed, throughput-intensive workloads	Lowest cost HDD volume designed for less frequently accessed workloads
Use Cases	I/O-intensive NoSQL and relational databases	Boot volumes, low-latency interactive apps, dev and test	Big data, data warehouses, log processing	Colder data requiring fewer scans per day
API Name	io1	gp2	st1	sc1
Volume Size	4 GB–16 TB	1 GB–16 TB	500 GB–16 TB	500 GB–16 TB
Max IOP	32,000	10,000	500	250

**S/V
olu
me**

Max 500 MB/s 160 MB/s 500 MB/s 250 MB/s
Thr
oug
hpu
t/Vo
lum
e

Max 80,000 80,000 80,000 80,000
IOP
S/In
stan
ce

Max 1,750 MB/s 1,750 MB/s 1,750 MB/s 1,750 MB/s
Thr
oug
hpu
t/Ins
tanc
e

Do IOPS IOPS MB/s MB/s
min
ant
Perf
orm

**ance
Attr
ibut
e**

A popular choice is the general-purpose SSD. A classic time to use this is when you want a boot volume for your virtual machine. This is an excellent mechanism for something that needs a bit higher performance.

You can go from a gigabyte to 16 terabytes with these. You get three IOPS per gigabyte, and it caps out at 10,000 IOPS. It's helpful to keep that in mind. So if you have a 1 TB general-purpose SSD, you will experience about 3,000 IOPS for that disk.

There is a credit system for bursting. So you can burst above those amounts, especially after periods of relatively little use. Once again, from a cost perspective, Amazon Web Services will charge you for the size of the SSD. It doesn't care how much data is actually on it.

If you need better performance, you could step up to a provisioned IOPS SSD. Now you're not only dictating the size from 4 GB to 16 TB, but you're also able to indicate your targeted IOPS up to a maximum of 20,000. If you need even more performance, the solution is to use one of these provisioned IOPS SSDs but combine that with RAID 0.

This capability allows you to have a bigger, higher performance SSD to avail yourself of Amazon Web Services. The cost model for the provision to IOPS SSD is a little

different, as you might guess, because Amazon Web Services will charge you based on the size and the IOPS that you're targeting.

You can use these provisioned IOPS SSDs for database storage, for example, where you need to guarantee great throughput. It probably goes without saying that things are continually changing in Amazon Web Services, and some new volume types are available.

Only time will tell what other types of EBS volumes that Amazon Web Services will make available. So you have an option to use an EBS volume that will be great for what you want to accomplish. Maybe you will be storing log files, a lot of them, and won't be accessing them all that frequently, and the right performance isn't that big of a deal. In that case, you could go with a cold HDD. However, if you have some high-performance EC2 instance that you want to boot from an EBS volume, you would go with a general-purpose SSD or maybe even step it up to some provisioned IOPS that you can add to the equation.

DESIGNING RESILIENT EFS SERVICES

You may have a need for a Network File System ([NFS](#)) compatible storage structure in the cloud. AWS Elastic File Service (EFS) is the AWS managed solution you could use. Like S3, EFS can be scaled easily as needs require.

To experiment with EFS, you should consider an Amazon Linux AMI because it has built-in NFS capabilities. This eliminates the need to install NFS separately. Be sure to permit NFS in the security group used by your EC2 instances that are

going to call on the EFS file system, however.

Lab: A Basic EFS Configuration

To implement EFS, follow these steps:



Step 1. Search for EFS in the Management Console and select it.

Step 2. Choose **Create File System** to open the window shown in [Figure 2-3](#).

Step 3. Select the VPC for your EFS file system from the list.

Step 4. Select the Availability Zones that you want AWS to create mount targets for.

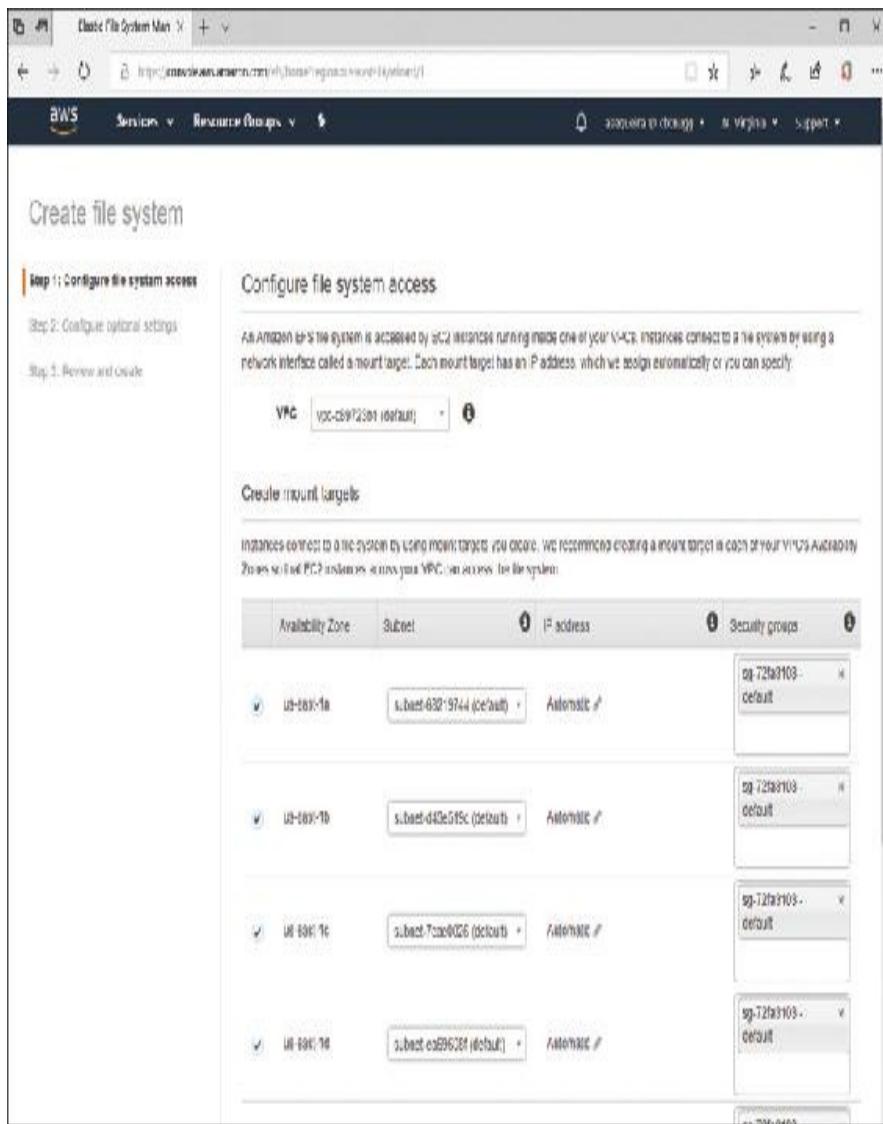
Step 5. Choose **Next Step**.

Step 6. Name your file system, choose your default performance mode, and choose **Next Step**.

Step 7. Choose **Create File System**.

Step 8. Choose your file system from the list and make a note of the File system ID value.

Figure 2-3 Creating a New EFS File System



To use the EFS file system from an Amazon EC2 instance, you mount the EFS file system. The following example uses an Amazon Linux EC2 instance because it has NFS preinstalled for you. Be sure to permit NFS in the security group used by your EC2 instance as well as the security group used for the file system. Follow these steps:

Step 1. Connect to your Amazon Linux EC2 instance using SSH.

Step 2. Install the `amazon-efs-utils` package, which has the Amazon EFS mount helper. To install this package,

use the following command:

```
sudo yum install -y amazon-efs-utils
```

Step 3. Make a directory for the mount point with the following command:

```
sudo mkdir myefs
```

Step 4. Mount the Amazon EFS file system to the directory that you created. Use the following command and replace file-system-id with your File System ID value:

```
sudo mount -t efs fs-12345678:/ ./myefs
```

Step 5. Change directories to the new directory that you created with the following command:

```
cd myefs
```

Step 6. Make a subdirectory and change the ownership of that subdirectory to your EC2 instance user. Then navigate to that new directory with the following commands:

```
sudo mkdir test  
sudo chown ec2-user test  
cd test
```

Step 7. Create a sample text file for storage in EFS with the following command:

```
touch test-file.txt
```

Remember, multiple EC2 instances across your different Availability Zones would also be able to use this file system. One of the compelling aspects of the EFS file system in AWS is that multiple nodes and users can access the file system simultaneously. This is why EFS is often used for shared storage access to common files. EFS also offers other potential benefits such as file sync options and the ability to connect/mount EFS file systems from on-premises via Direct Connect.

Lab Cleanup

Step 1. In AWS, choose **File Systems** from the left column in the Management Console for EFS.

Step 2. Select your file system and choose **Delete File System** from the Actions drop-down menu.

Step 3. Enter your File System ID (copy the value displayed just above) and click **Delete File System**.

DESIGNING RESILIENT GLACIER SERVICES



AWS Glacier is technically part of S3, but you should note there are many fundamental differences. Sure, it gives us 11 nines of durability like the other classes of S3, but there are more differences than similarities. Here are some of the critical unique characteristics you need to be aware of:

- It is specially designed for backup and archiving.
- ZIPs and TARs that you store in Glacier can be huge—up to 40 TBs in size, so here you shatter size limitations of other classes.
- You will be charged more if you find that you are accessing Glacier data more frequently.
- You create Vaults inside Glacier for storage.
- You can create 1,000 Vaults per AWS root account, but this number can be negotiated with Amazon.
- The Archive ID (naming) of stored objects is automated by AWS.
- Like S3, security compliance is an option for a variety of different regulations and standards.
- Amazon does not provide a GUI for the transfer of objects in and out of your Vaults.
- You use the CLI or an SDK or a Lifecycle Policy to transfer objects.

Lab: Creating a Vault

In this lab, you see how to create a Glacier Vault in AWS.



Step 1. Search in the AWS Management Console for Glacier.

Step 2. Choose the appropriate region to work with.

Because Glacier is region specific and not a global resource like S3, you do not need to worry about unique naming across all of AWS.

Step 3. Select **Create Vault** to open the window shown in Figure 2-4.

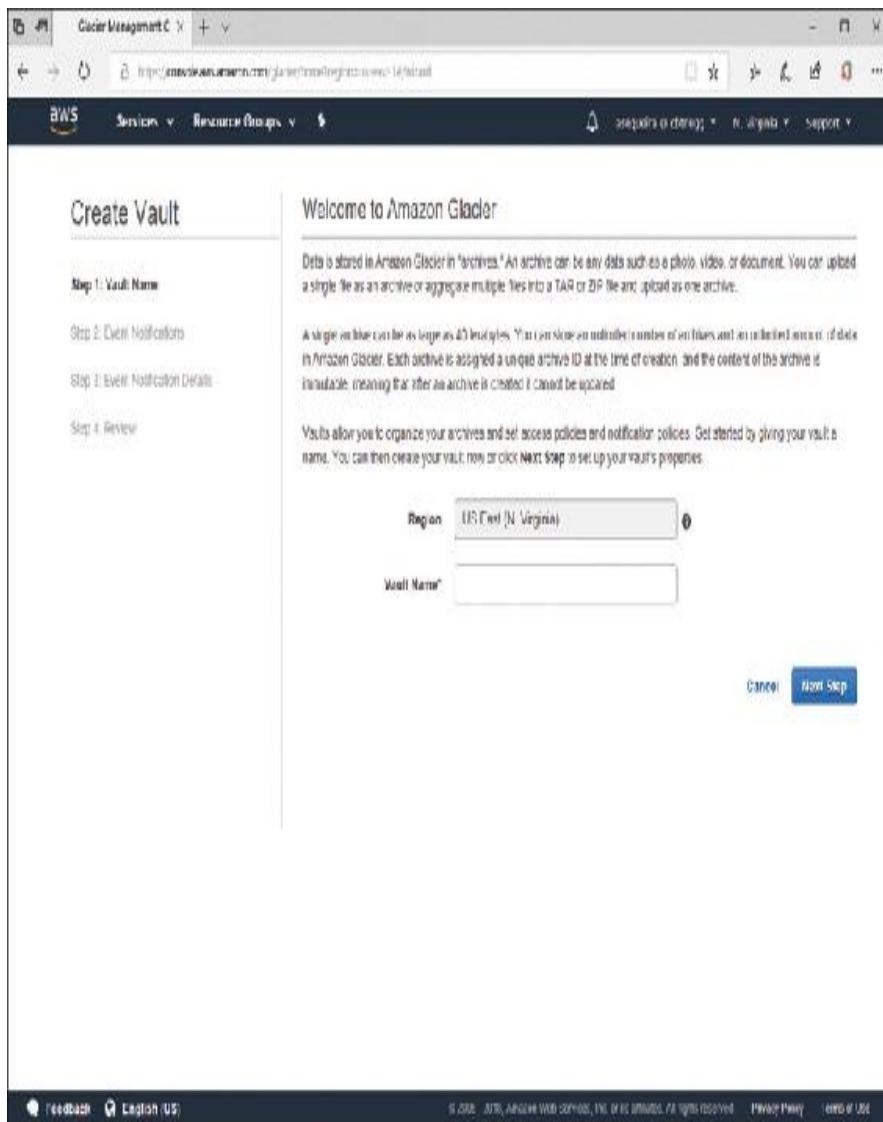


Figure 2-4 Creating an AWS Glacier Vault

Step 4. Select your Region (if you failed to initially in your Management Console) and give it a Vault name. Click **Next**.

Step 5. Notice you can set event notifications. This will work with Amazon's Simple Notification Service (SNS). You can enable notifications with a new SNS service topic or enable notifications using an existing SNS topic that you created. Click **Next**

Step.

Step 6. After reviewing your Glacier settings, click **Submit**.

Lab Cleanup

Step 1. In the Amazon Glacier Vaults console, highlight your Vault.

Step 2. Choose the **Delete Vault** button.

Step 3. Click **Delete Vault** in the warning window.

If you go to the settings of the Vault, you can create a retrieval policy. You could configure a retrieval policy that permits you to deny retrieval requests that might, say, violate the free tier account policy and cause charges. You can also set up a max retrieval rate, configure a setting to allow 1 GB per hour, or eliminate any retrieval rate. You can also efficiently provision capacity. This helps guarantee more expedited retrievals when needed.

Remember, no GUI is provided for object transfer. You can go to the CLI and upload data to your Amazon Glacier vault, or you could upload data using an SDK.

For example, you might use the following CLI syntax to upload a test.zip archive into a Vault named Test2:

```
aws glacier upload-archive --account-id - - -
vault-name Test2 --body
test.zip
```

Note

I use a dash (-) for the account ID to indicate to use the account

currently logged in at the CLI.

Running this command causes AWS to upload the object and display the archive ID that is generated. A checksum is also generated. Moreover, the exact location of your archive in the vault is returned to you.

Note

Remember, there is always latency when working with Glacier. So, for example, it might take some time to see the archive objects in your Vault inside the GUI.

EXAM PREPARATION TASKS

As mentioned in the section “How to Use This Book” in the Introduction, you have a couple of choices for exam preparation: the exercises here, [Chapter 16, “Final Preparation,”](#) and the exam simulation questions in the Pearson Test Prep practice test software.

REVIEW ALL KEY TOPICS

Review the most important topics in this chapter, noted with the Key Topics icon in the margin of the page. [Table 2-4](#) lists a reference of these key topics and the page numbers on which each is found.



Table 2-4 Key Topics for Chapter 2

Key Topic Element	Description	Page Number
List	S3 Features	42
List	S3 Storage Classes	42
Steps	Create an S3 Bucket	44
Steps	Create an EBS Volume	48
Section	Elastic Block Store	49
Steps	Implement EFS	51
Section	Designing Resilient Glacier Services	53
Steps	Create a Glacier Vault	54

COMPLETE TABLES AND LISTS FROM MEMO DV

MEMORY

Print a copy of Appendix C, “Memory Tables” (found on the book website), or at least the section for this chapter, and complete the tables and lists from memory. Appendix D, “Memory Tables Answer Key,” also on the website, includes completed tables and lists to check your work.

DEFINE KEY TERMS

Define the following key terms from this chapter and check your answers in the glossary:

Durability

Object Storage

Storage Classes

Bucket

Instance Store

NFS

Vault

Q&A

The answers to these questions appear in Appendix A. For more practice with exam format questions, use the Pearson Test Prep practice test software.

1. What are the four storage classes of AWS S3?
2. What is the durability of AWS S3?
3. What is the minimum number of AZs that S3 will use with the standard class?
4. What does it mean when EBS is described as persistent?

5. What protocol is used to access the data stored in the Amazon EFS service?

6. Glacier is designed for what type of storage?

Chapter 3. Designing Decoupling Mechanisms

This chapter covers the following subjects:

- **Decoupling Demystified:** What exactly does it mean to design decoupling mechanisms in AWS? This section makes sense of this concept for you if you’re not familiar with it.
- **Advantages of Decoupled Designs:** What advantages can you gain if you design decoupled services and resources? This section provides many of the possible advantages.
- **Synchronous Decoupling:** One type of decoupling is called synchronous decoupling. This section details this approach.
- **Asynchronous Decoupling:** Another type of decoupling is termed asynchronous. This section examines decoupling in detail, and you learn how to configure the AWS SQS service, which is often a key element in the desired “loose” coupling architectural approach.

Are you ready to learn about what most consider to be a key architectural best practice in AWS? This chapter ensures you know about decoupling in the scope of AWS as well as a key service that can accommodate it—the Simple Queue Service (SQS).

“DO I KNOW THIS ALREADY?” QUIZ

The “Do I Know This Already?” quiz allows you to assess whether you should read the entire chapter. Table 3-1 lists the major headings in this chapter and the “Do I Know This Already?” quiz questions covering the material in those headings so you can assess your knowledge of these specific areas. The answers to the “Do I Know This Already?” quiz appear in [Appendix A](#).

Table 3-1 “Do I Know This Already?” Foundation Topics
Section-to-Question Mapping

Foundation Topics Section	Questions
Decoupling Demystified	1
Advantages of Decoupled Designs	2
Synchronous Decoupling	3
Asynchronous Decoupling	4–5

Caution

The goal of self-assessment is to gauge your mastery of the topics in this chapter. If you do not know the answer to a question or are only partially sure of the answer, you should mark that question as wrong for purposes of the self-assessment. Giving yourself credit for an answer you correctly guess skews your self-assessment results and might provide you with a false sense of security.

- 1.** What two aspects of decoupling are often desired when architecting AWS solutions? (Choose two.)
 - a.** Dependent
 - b.** Autonomous

c. Tight

d. Loose

2. Which of the following is not considered an advantage of decoupled designs?

- a. To ensure failures in one component have a minimal effect on others
- b. To have well-defined interfaces that allow component interaction
- c. The promotion of new components being launched at any time
- d. To reduce the required messaging between application components

3. What is the key characteristic of a synchronous decoupling in an AWS design?

- a. The decoupled components must always be available.
- b. The decoupled components do not always need to be present.
- c. The decoupled resources must be of the compute category.
- d. The decoupled components must use FIFO queues in their messaging between each other.

4. What services of AWS might promote asynchronous decoupling directly? (Choose two.)

- a. Neptune
- b. SQS
- c. CloudWatch

d. Kinesis

5. What types of message queues can you create in SQS?

(Choose two.)

- a.** Standard
- b.** Advanced
- c.** LIFO
- d.** FIFO

FOUNDATION TOPICS

DECOUPLING DEMYSTIFIED



As you should know, decoupling is a big deal in the new blueprint for the AWS Certified Solutions Architect Associate level exam. This section outlines some of the highlights of this interesting term and approach.

First of all, what does *decoupling* even mean? This term refers to components remaining potentially autonomous and unaware of each other as they complete their work for some greater output. This decoupling can be used to describe components that make up a simple application, and the term even applies on a broad scale. For example, many point to the fact that in public cloud computing, the architecture is decoupled in that another entity like Amazon will completely handle the physical infrastructure of the network for you, while you work with the data and applications hosted on this hardware. Note that you are not informed of the exact operations and

procedures carried out by Amazon, while it is not entirely positive what you are doing with the resources you are provisioning from a high-level business goal perspective. Obviously, Amazon can view and monitor your overall resource deployments.

The Solutions Architect exam focuses on decoupling your applications in the AWS ecosystem. You are encouraged to break up your application into smaller, loosely coupled components. As you'll read in the next section, this has the advantages of permitting you to reduce interdependencies and reduce the impact that one failed component can have on the entire system of application components and thus the application itself.

Note

We typically refer to the components that are not fully dependent on other components as *black boxes*.

How can the components interact in a system like this? They can use specific, technology-agnostic interfaces (such as with RESTful APIs). Thus, components can be easily modified, and as long as the component maintains backward compatibility, it can still function just fine with the other components of the system.

Note

You might take advantage of the AWS API Gateway in such a design. This fully managed service makes it easier for you to publish, maintain, monitor, and secure APIs to any scale you require.

ADVANTAGES OF DECOUPLED

DESIGNS

As described in the preceding section, decoupling reduces interdependencies. If you are following best practices, these decoupled resources are loosely coupled. What are the major advantages of such decoupled designs?

Here are just some:



- To help ensure changes or failures in one component have a minimal effect on others.
- To have well-defined interfaces that allow the various components to interact with each other only through specific, technology-agnostic interfaces. The ability to modify any underlying operations without affecting other components should be made possible.
- To enable smaller services to be consumed without prior knowledge of their network topology details through loose coupling. This enables you to launch or terminate new components at any point.
- To reduce the impact on the end users and increase your ability to make progress on your offline procedures.

SYNCHRONOUS DECOUPLING



For the exam, you need to be able to distinguish between two valid decoupling techniques in AWS: synchronous decoupling and asynchronous decoupling.

With synchronous decoupling, you have two components (for example) that must always be available for the overall resource to function properly. Although they both must always be available, they certainly are “unaware” of each other, which

means they are truly decoupled. Understand here that the term *components* can represent many different things with decoupling. It might refer to an EC2 instance or the entire EC2 service itself.

An example of synchronous decoupling in AWS is using Elastic Load Balancing (ELB) to distribute traffic between EC2 instances that are hosted in multiple Availability Zones. For this to function properly, you need at least one EC2 instance in each AZ. They are unaware of each other, but they both had better be there; otherwise, you have no load balancing. What is great is the fact that you can add nodes to this configuration and even later remove them without disrupting anything!

ASYNCHRONOUS DECOUPLING

With asynchronous decoupling, communication can still be achieved between the components even if one of the components is temporarily unavailable.

An example of asynchronous decoupling is using the Simple Queue Service (SQS) to handle messaging between components. You can have a component temporarily go offline, and the message can be properly queued until the component is back online.

When you use this approach, one component generates events while another component consumes them. They are not communicating directly but through a service like SQS or perhaps even a streaming data platform like AWS Kinesis.

With SQS, additional resiliency is achieved; if a process that is reading messages from the queue fails, messages can still flow

to the queue, and the recovered process or even another process can pick up the messages that have been waiting. You might even have a less scalable back-end server that requires this decoupling so that it can handle bursts in demand for its resources.

Here is an example of a loosely coupled asynchronous application architecture in practice. Suppose you run a website business that permits videos to be uploaded and you then provide them to users with closed captioning in the desired language. The system might operate as follows:

1. 1. The website automates the upload of the original video to an S3 bucket.
2. 2. A request message is sent to an SQS queue. The message contains a pointer to the video and the target closed captioning language.
3. 3. The service for completing the closed captioning runs on an EC2 instance. That service reads the message and retrieves the correct video and learns the desired language contained in the SQS message.
4. 4. The new video is placed in another S3 bucket, and the service running in EC2 generates a response message in the appropriate queue created in SQS. This message points to the location of the new video.

In the preceding system, note that you can monitor the SQS queue depth and spin up EC2 instances as needed to accommodate the required workload.

Lab: Configure SQS

Let's examine SQS inside AWS with a sample configuration:



Step 1. Using the drop-down menu option at the top of the AWS Management Console, ensure you are in the

correct Region. You will create a FIFO queue that is available only in select Regions, such as the US East (N. Virginia), US East (Ohio), US West (Oregon), and EU (Ireland) Regions. FIFO queues preserve the order in which messages are sent and received.

Step 2. Sign in to the AWS Management Console and search for **SQS**.

Step 3. Choose **Get Started Now**. This screen appears if you have not yet configured SQS queues. Otherwise, choose **Create New Queue**.

Step 4. On the Create New Queue page, for Queue Name, type **SQS_Sample fifo** and select **FIFO Queue**. Then select the **Configure Queue** button.

Step 5. Hover your mouse over the Information icons to learn the meaning of these various configuration parameters, as shown in Figure 3-1.

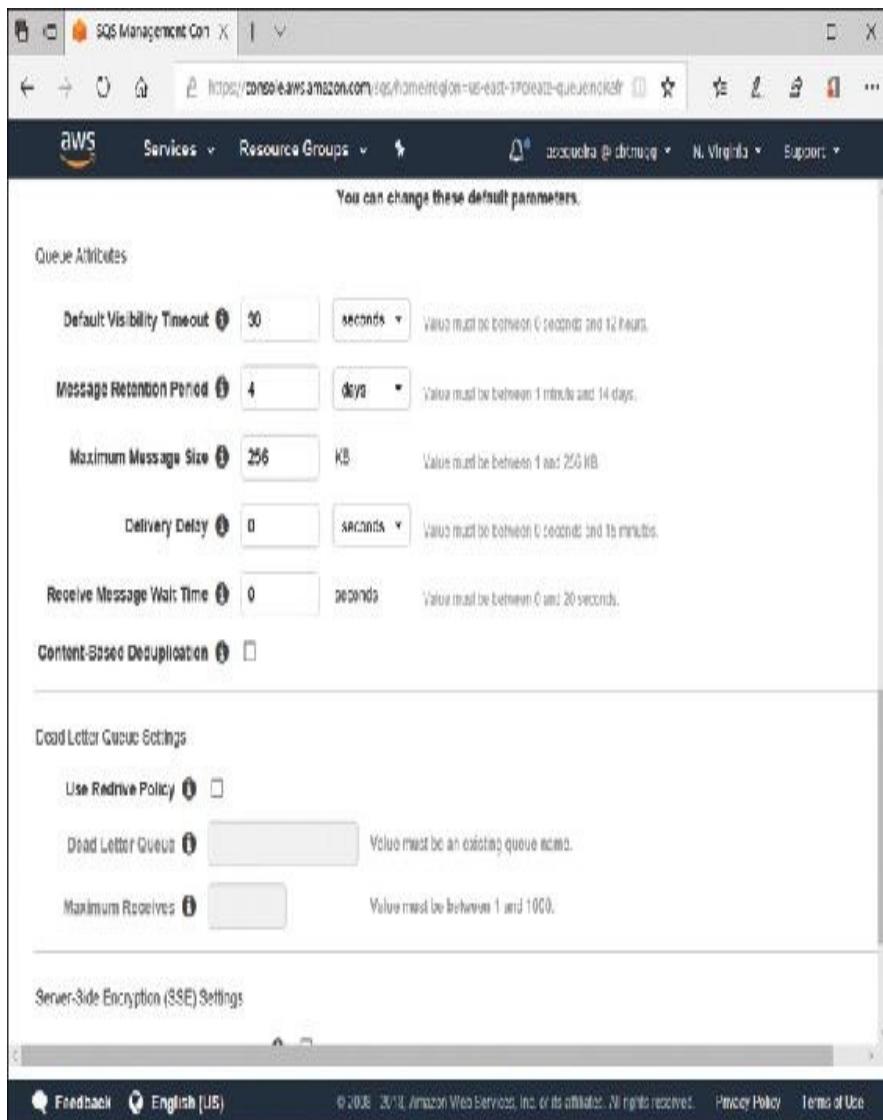


Figure 3-1 Configuring the SQS Queue

Step 6. Change the Default Visibility Timeout value to **40** seconds. This makes the message invisible to other potential nodes for 40 seconds after the message has been retrieved. Remember, *retrieved* in this case means the consumer or processor obtained the message from the queue.

Step 7. Click **Create Queue**. You see your queue listed in the console, as shown in Figure 3-2.

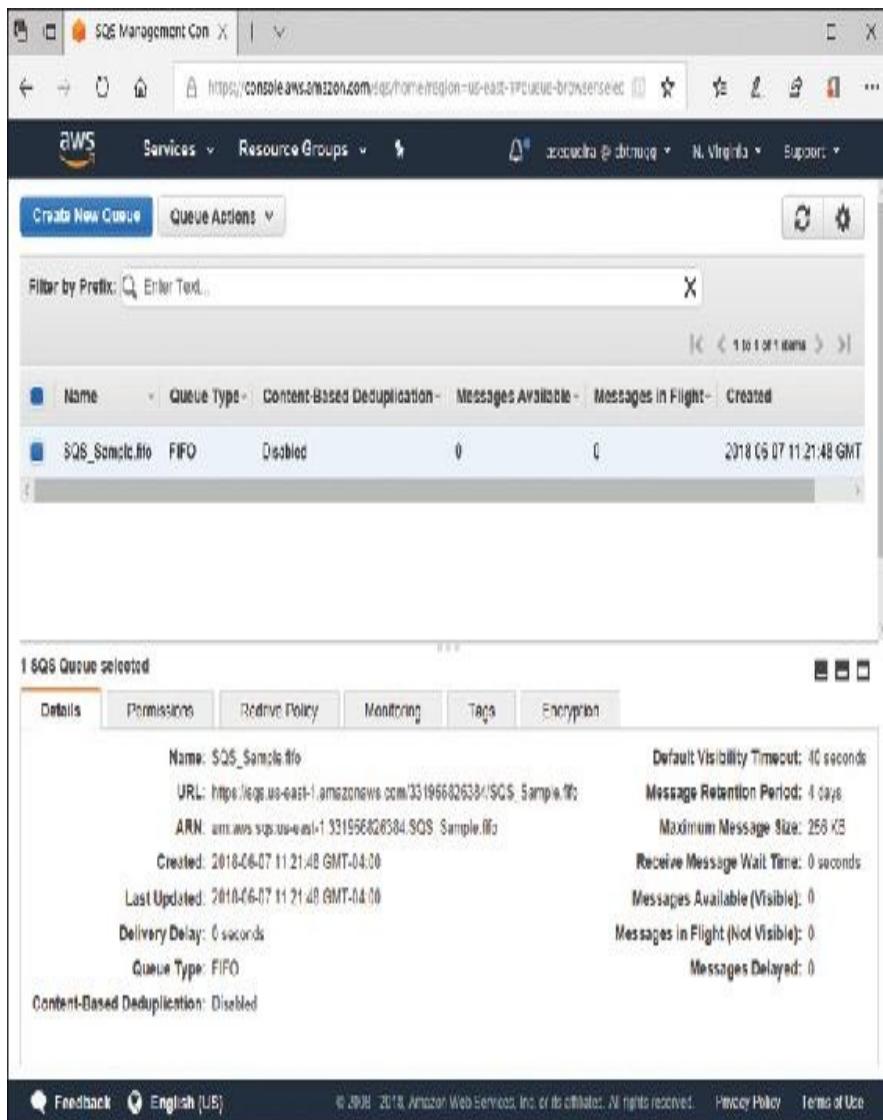


Figure 3-2 Examining your Queue in SQS

Step 8. To create a sample message, select the **Queue Actions** drop-down button and choose **Send a Message**.

Step 9. To send a message to a FIFO queue, type **This is my sample message text** in the Message Body. In the Message Group ID, type **MyGroupId1234**, and in the Message Deduplication ID field, type **MyDedupId1234**. Then choose the **Message**

Attributes tab.

Step 10. For the Message Attribute Name, type **MySampleAttribute**. For the Message Attribute Data Type, select **Number** and type **byte** for the optional custom type. For the Message Attribute Value, type **24**, as shown in Figure 3-3.

Step 11. Choose **Add Attribute**.

Step 12. Click **Send Message**.

Step 13. Click **Close**.

Send a Message to SQS_Sample.fifo X

Message Body Message Attributes

Name	MySampleAttribute
Type	Number ▼ byte
Value	24 X

Enter a numerical value.

Add Attribute [What is Message Attribute?](#)

Name	Type	Values

Cancel Send Message

Figure 3-3 Configuring a Message Attribute

Lab Cleanup

Step 1. Ensure you are in the SQS service of your AWS Management Console.

Step 2. Select the queue you created in this lab.

Step 3. Choose **Delete Queue** from the Queue Actions drop-down menu.

Step 4. Choose Yes, Delete Queue.

EXAM PREPARATION TASKS

As mentioned in the section “How to Use This Book” in the Introduction, you have a couple of choices for exam preparation: the exercises here, Chapter 16, “Final Preparation,” and the exam simulation questions in the Pearson Test Prep practice test software.

REVIEW ALL KEY TOPICS

Review the most important topics in this chapter, noted with the Key Topics icon in the margin of the page. Table 3-2 lists a reference of these key topics and the page numbers on which each is found.

Table 3-2 Key Topics for Chapter 3

Key Topic		
Key Topic Element	Description	Page Number
Overview	Decoupling demystified	61
List	Advantages of decoupled designs	62

Overview	Synchronous decoupling	62
Steps	Configuring SQS	63

COMPLETE TABLES AND LISTS FROM MEMORY

There are no memory tables in this chapter.

DEFINE KEY TERMS

Define the following key terms from this chapter and check your answers in the glossary:

Decoupling

Loose decoupling

Synchronous decoupling

Asynchronous decoupling

Q&A

The answers to these questions appear in Appendix A. For more practice with exam format questions, use the Pearson Test Prep practice test software.

1. Describe decoupled architectures.
2. Provide two advantages to decoupled designs.
3. Describe synchronous decoupling.
4. Describe asynchronous decoupling.

5. Name the messaging service of AWS that promotes loosely coupled components.

Chapter 4. Designing a Multitier Infrastructure

This chapter covers the following subjects:

- ■ **Single-Tier Architectures:** Need a simple and straightforward approach to a technology architecture? It doesn't get much simpler than the single-tier architecture. This section describes this approach and discusses the advantages of this design.
- ■ **Multitier Architectures:** So much has been made about the wonders of a multitier architecture. This section discusses this approach in great detail and provides examples of robust multitier architectures in AWS.

This chapter examines both single-tier and multitier architectures with a focus on AWS (of course). You will leave this chapter well versed on the differences between these two approaches, and you will also understand how they could easily be implemented in an AWS environment.

“DO I KNOW THIS ALREADY?” QUIZ

The “Do I Know This Already?” quiz allows you to assess whether you should read the entire chapter. Table 4-1 lists the major headings in this chapter and the “Do I Know This Already?” quiz questions covering the material in those headings so you can assess your knowledge of these specific areas. The answers to the “Do I Know This Already?” quiz appear in [Appendix A](#).

Table 4-1 “Do I Know This Already?” Foundation Topics
Section-to-Question Mapping

Foundation Topics Section	Questions
Single-Tier Architectures	1–2
Multitier Architectures	3–4

Caution

The goal of self-assessment is to gauge your mastery of the topics in this chapter. If you do not know the answer to a question or are only partially sure of the answer, you should mark that question as wrong for purposes of the self-assessment. Giving yourself credit for an answer you correctly guess skews your self-assessment results and might provide you with a false sense of security.

- 1.** What is the AWS service that is the main focus of a typical single-tier architecture in AWS?
 - a.** Cognito
 - b.** Redshift
 - c.** EC2
 - d.** Kinesis

- 2.** What is often considered a major advantage of a single-tier architecture in AWS?
 - a.** Scalability
 - b.** Fine-grained management
 - c.** Lower costs
 - d.** Enhanced security

3. Which of the following is not a common tier in a multitier architecture?
- Presentation
 - Identity
 - Logic
 - Data
4. What is an advantageous logic tier option in AWS that provides serverless compute?
- Kinesis
 - ElastiCache
 - Lambda
 - Redshift

FOUNDATION TOPICS

SINGLE-TIER ARCHITECTURES

A *single-tier architecture* is often called a *one-tier architecture*. No matter what you call it, it is a simple architecture in which all the required components for a software application or technology are provided on a single server or platform. For example, you might construct a web-based application in AWS that utilizes a single EC2 instance for all the required elements. This might include:

- A web server
- The end-user interface
- Middleware required to run the logic of the application (beyond what the underlying OS can provide)

- Back-end data required for the application

Figure 4-1 shows an example of a single-tier architecture in AWS. It has a single, large EC2 instance that runs Windows Server 2016. This single-tier architecture calls upon SQL Server 2016 for the application code and database functionality and Internet Information Server (IIS) for the web server and end-user interface code.



Figure 4-1 A Single-Tier Architecture Example

In Figure 4-1, you could argue that “tiers” inside the EC2 instance provide the application functionality. This is certainly true because the database work is handled by SQL Server 2016 and the web server by IIS. Some application architectures would not even have this level of tiering. They might have an application that runs within an OS that provides all the services needed thanks to programming inside the single application. These single-tier architectures might prove even more difficult to improve upon and troubleshoot.

While you might sneer at the single-tier application as too basic, this design is appropriate for many circumstances and does offer potential advantages. Here are just a few:



- It is simple to understand and design.

- It is easy to deploy.
- It is easy to administer.
- It has potentially lower costs associated with it.
- It can be easy to test.
- It is often ideal for environments that do not require scalability.

Lab: Building a Single-Tier Architecture with EC2

This lab demonstrates how simple it is to spin up a single-tier architecture server operating system in EC2:



Step 1. Search the AWS services for EC2 and select the link to enter the EC2 Dashboard.

Step 2. Select **Launch Instance** to create a new EC2 instance for your single-tier architecture.

Step 3. Select the AMI for your architecture. For this lab, choose the **Microsoft Windows Server 2016 Base with Containers** AMI. Notice that this image is Free-Tier Eligible.

Note

Containers are an OS-level virtualization method for deploying and running distributed applications without launching an entire VM for your application. Interestingly, containers can make a multitier architecture possible on a single instance. If you have each “tier” in its own container, running on a single instance, it would be easy to argue that a true multitier architecture is in place. This is very different from simply having multiple tiers installed on a single instance directly. The container approach enables the tiers to be very easily moved to other instances as growth occurs.

Step 4. Click **Next: Configure Instance Details**.

Step 5. Review the defaults for your EC2 instance, as shown in Figure 4-2. Click **Next: Add Storage** when you’re finished with your review.

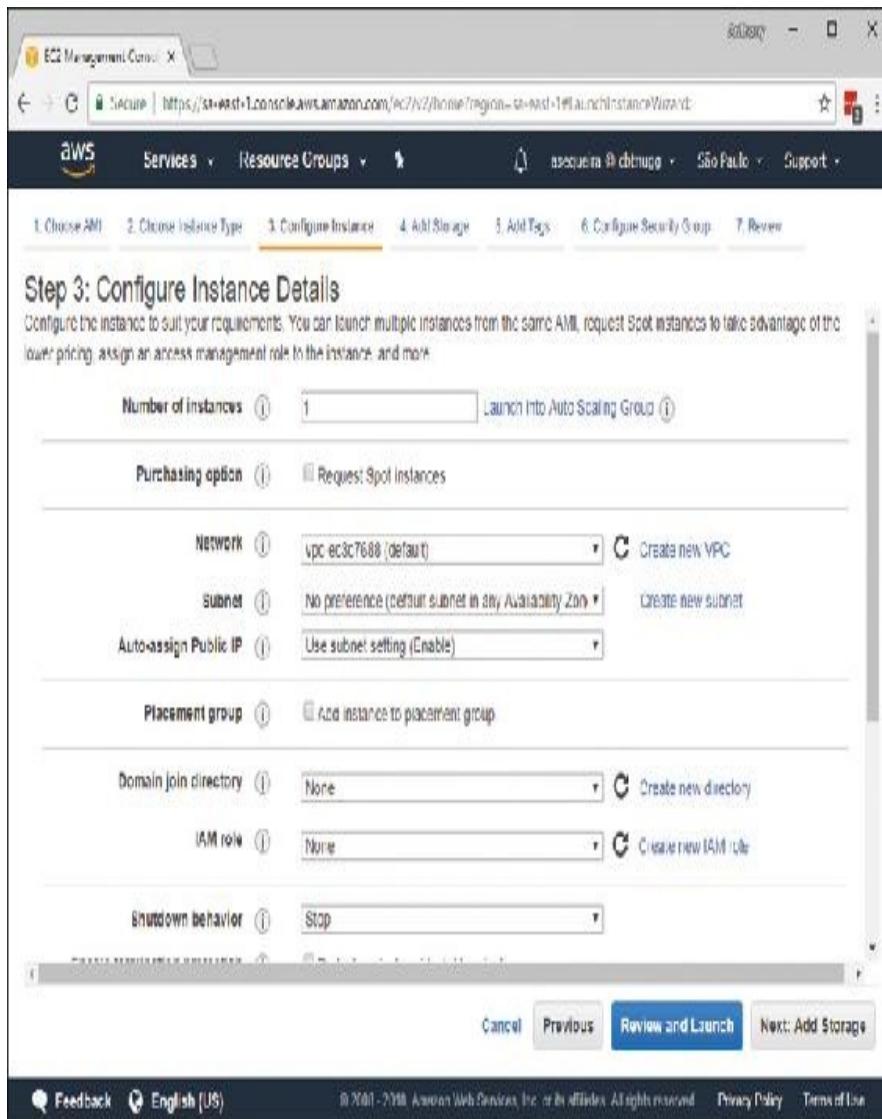


Figure 4-2 Configure the EC2 Instance Details

Step 6. Review the default settings for your EC2 storage and click **Next: Add Tags** when you’re done with your review.

Step 7. Click **Add Tag** and enter **Name** in the key field. In

the Value field, enter **TESTARCH**. Click **Next: Configure Security Group** when done.

Step 8. Click **Add Rule** in the Security Group page and choose **HTTP** from the Type drop-down menu. Click **Review and Launch**.

Step 9. Click **Launch**.

Step 10. In the Key Pair window, choose **Create a New Key Pair** from the drop-down menu. For Key Pair Name, enter **TESTARCH** and click **Download Key Pair**. Click **Launch Instances**.

Step 11. Scroll down and choose **View Instances** to view your running instance for your single-tier architecture.

Lab Cleanup

Step 1. In the EC2 dashboard, click the link to view your running instances.

Step 2. Select the instance you created in this lab and choose **Actions > Instance State > Terminate**.

Step 3. Click **Yes, Terminate**.

Step 4. In the left column, choose **Key Pairs** under the Network & Security grouping.

Step 5. Select the key pair you created for this lab and choose **Delete**. Click **Yes**.

Step 6. Also under the Network & Security grouping, choose **Security Groups**.

Step 7. Select the **Launch-Wizard** security group from

this lab and choose **Delete Security Group** from the Actions menu. Choose **Yes, Delete**.

Note

If your instance is not yet terminated, you will receive an error when you attempt to delete the security group. Be sure to wait enough time for the instance to terminate.

MULTITIER ARCHITECTURES

People have been raving about the *multitier architecture* for decades now. Keep in mind that you will often see it described slightly differently. For example, some call it a *three-tier architecture* (if it truly has three tiers) or an *n-tier architecture* (if there are more than three layers).

Many multitier architectures are built using a service-oriented architecture (SOA) using various web services. Thanks to AWS, you can leverage the API Gateway and AWS Lambda to assist with the execution of code that can help integrate tiers with other tiers. The API Gateway service helps you create and manage the APIs, and the Lambda service enables the execution of arbitrary code functions.

AWS Lambda even includes the capability to create functions that communicate within your Virtual Private Cloud (VPC). This permits you to design multitier architectures without violating the requirements of network privacy. For example, you can integrate web services with relational database services that contain highly sensitive, private information.

There are many advantages discovered through the use of a multitier design. They can include:

Key Topic

- ■ Improved security
- ■ Improved performance
- ■ Improved scalability
- ■ Fine-grained management
- ■ Fine-grained maintenance
- ■ Higher availability

Multitier design also:

- ■ Promotes decoupling of components
- ■ Promotes multiple teams for architecture development, implementation, and maintenance

Table 4-2 shows sample components of AWS that might be used for a multitier architecture and the requirements they meet.

Table 4-2 Sample Components of an AWS Multitier Architecture

Requirement	Service
DNS Resolution	Route 53
Content Delivery Network	CloudFront

Web Resources	Simple Storage Service
Web Servers	Elastic Compute Cloud
Load Balancing	Elastic Load Balancing
Scalability	Auto Scaling
Application Servers	Elastic Compute Cloud
Database Servers	Relational Database Services

In this example, note the following:

- DNS resolution of client requests is handled by Route 53. This service helps route traffic requests to and from clients to the correct AWS services for the application and to the correct internal infrastructure components within the AWS data centers. Route 53 features vast scalability and high availability for this task.
- Content from your application (streaming, static, and/or dynamic) can be delivered through CloudFront. CloudFront provides an impressive global network of edge locations. CloudFront can help ensure that client requests are automatically routed to the nearest edge location so that content can be delivered with the lowest possible latency.
- S3 can provide valuable object-based storage for your multilayer architecture. This might even include the static content required by any web

content required for your solution.

- ■ Elastic Load Balancing can handle HTTP requests to distribute them between multiple EC2 instances. Note that these EC2 instances can be strategically located across different Availability Zones to increase availability. Also note that the ELB service can dramatically increase the fault tolerance of your solution.
- ■ EC2 instances can function as the web servers in a web-based application. An AMI is chosen that facilitates such web services.
- ■ An Auto Scaling group can be used for the web servers to facilitate scalability. This allows the automatic expansion of instances when requests spike and can even eliminate instances when they are not needed to help save on costs.
- ■ Amazon RDS can provide the data required for the application. To help with high availability, the database servers can be located in different Availability Zones. Synchronous replication can be used between these servers to ensure accurate data in all of them.

THE CLASSIC THREE-TIER ARCHITECTURE



Figure 4-3 shows the elements of a classic three-tier architecture.

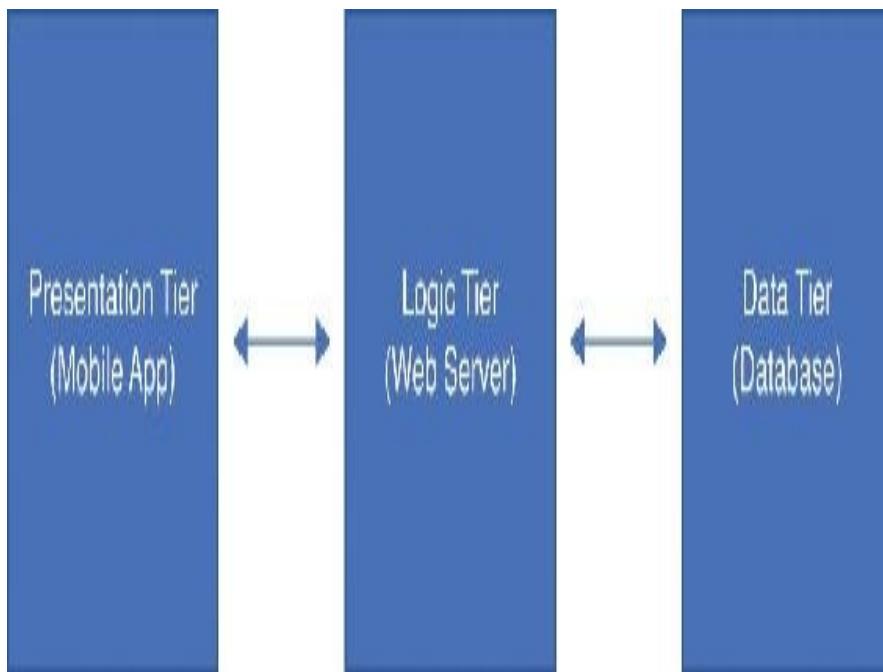


Figure 4-3 The Three-Tier Architecture

The three-tier architecture consists of the following:

- **Presentation tier:** Consists of the components that users interact with, which might include web pages and/or mobile app user interface components
- **Logic tier:** Contains the code required to translate user actions initiated at the presentation tier to the functionality required by the application
- **Data tier:** Consists of storage media that hold the data relevant for the application architecture, which might be database, object stores, caches, or file systems

AWS offers many technologies to assist with the presentation tier of your architecture. The Amazon Cognito service can assist with the creation and management of user identities. S3 might be the source of any static web content you need to store and then deliver for the tier. The API Gateway can be enabled for cross-origin resource sharing compliance. This permits web browsers to invoke APIs from within your static web pages.

You can consider the logic tier to be the brains of the architecture. This certainly lends itself to the power of the API Gateway and Lambda functionality. They combine to allow a revolutionary new serverless approach to the multitier architecture. Thanks to advantages that come with serverless implementations, new levels of high availability, scalability, and security are possible. While a more traditional server-based implementation of the three-tier architecture might require thousands of servers in order to scale, a serverless environment does not need a single virtual server in its operation.

The data tier of your architecture can be used with a wide variety of AWS solutions. They are often organized into two broad categories: Amazon VPC-hosted data stores and IAM-enabled data stores. VPC-hosted data store options include:

- ■ **RDS:** Provides several relational database engine options
- ■ **ElastiCache:** Boosts performance with in-memory caching
- ■ **Redshift:** Provides simple data warehousing capabilities
- ■ **EC2:** Provides data stored within technology offered by an EC2 instance(s)

IAM-enabled options include:

- ■ **DynamoDB:** An infinitely scalable NoSQL database
- ■ **S3:** Infinitely scalable object-based storage
- ■ **Elasticsearch Service:** A managed version of the popular search and analytics engine called Elasticsearch

EXAM PREPARATION TASKS

As mentioned in the section “How to Use This Book” in the

Introduction, you have a couple of choices for exam preparation: the exercises here, Chapter 16, “Final Preparation,” and the exam simulation questions in the Pearson Test Prep practice test software.

REVIEW ALL KEY TOPICS

Review the most important topics in this chapter, noted with the Key Topics icon in the margin of the page. Table 4-3 lists a reference of these key topics and the page numbers on which each is found.

Table 4-3 Key Topics for Chapter 4

Key Topic		
Key Topic Element	Description	Page Number
List	Advantages of the single-tier architecture	72
Steps	Building a single-tier architecture with EC2	72
List	Advantages of the multitier architecture	75

COMPLETE TABLES AND LISTS FROM MEMORY

Print a copy of Appendix C, “Memory Tables” (found on the book website), or at least the section for this chapter, and complete the tables and lists from memory. Appendix D, “Memory Tables Answer Key,” also on the website, includes completed tables and lists to check your work.

DEFINE KEY TERMS

Define the following key terms from this chapter and check your answers in the glossary:

Single-Tier Architecture

Multitier Architecture

Presentation Tier

Logic Tier

Data Tier

Q&A

The answers to these questions appear in Appendix A. For more practice with exam format questions, use the Pearson Test Prep practice test software.

1. Name at least four potential advantages of a single-tier architecture.
2. Name at least four potential advantages of a multitier

architecture.

- 3.** Name the three layers of the classic three-tier architecture.

- 4.** Name three AWS technologies that might be used for each layer of the classic three-tier architecture.

Chapter 5. Designing High Availability Architectures

This chapter covers the following subjects:

- **■ High Availability Compute:** Because compute is such a critical component of most AWS solutions, improving high availability for these resources is often desired. This section describes the various options for you.
- **■ High Availability Application Services:** Services that are important for applications should offer high levels of high availability. This section discusses high availability solutions for application services.
- **■ High Availability Database Services:** The database is often the foundation for data and AWS solutions. There are many options for HA solutions as described in this section.
- **■ High Availability Networking Services:** Because successful cloud IT services require reliable connectivity internal to the cloud network as well as from internal to external connectivity, HA in networking services is very important. This section discusses HA and networking.
- **■ High Availability Storage Services:** Today, more and more data is expected to be stored digitally, and more and more is expected to be stored in the cloud. This section discusses HA for your cloud storage.
- **■ High Availability Security Services:** It has been scientifically proven that if your security is not available, then it is not effective. This section covers HA for security.
- **■ High Availability Monitoring Services:** Monitoring needs to be there when required; otherwise, you cannot receive an accurate picture of performance and other key metrics. This section covers HA for monitoring your AWS infrastructure.

Remember, a compelling reason to consider cloud technologies is the ease with which you can improve upon (typically) your high availability (HA). This chapter discusses the design of HA architectures across various service areas of

AWS including compute, databases, monitoring, and more.

“DO I KNOW THIS ALREADY?” QUIZ

The “Do I Know This Already?” quiz allows you to assess whether you should read the entire chapter. Table 5-1 lists the major headings in this chapter and the “Do I Know This Already?” quiz questions covering the material in those headings so you can assess your knowledge of these specific areas. The answers to the “Do I Know This Already?” quiz appear in Appendix A.

Table 5-1 “Do I Know This Already?” Foundation Topics Section-to-Question Mapping

Foundation Topics Section	Questions
High Availability Compute	1
High Availability Application Services	2
High Availability Database Services	3
High Availability Networking Services	4

High Availability Storage Services	5
High Availability Security Services	6
High Availability Monitoring Services	7

Caution

The goal of self-assessment is to gauge your mastery of the topics in this chapter. If you do not know the answer to a question or are only partially sure of the answer, you should mark that question as wrong for purposes of the self-assessment. Giving yourself credit for an answer you correctly guess skews your self-assessment results and might provide you with a false sense of security.

- 1.** What addressing technology in AWS permits you to use a public network address for a resource and change the association of this address whenever you need to?
 - a.** Elastic IP Address
 - b.** Route 53 Pool
 - c.** VPN Endpoint Address
 - d.** Firewall NAT Point

- 2.** What feature in AWS permits the distribution of client requests to multiple AWS resources in a VPC?
 - a.** VPC Load Distribution services
 - b.** VPC Sharing
 - c.** Elastic Load Balancing

d. Auto Scaling

3. What RDS API action would show the AZ of the standby replica?

- a.** ListDBInstances
- b.** GetDBInstances
- c.** ShowDBInstances
- d.** DescribeDBInstances

4. When designing highly available connections to your AWS resources, what type of routing should you consider a priority?

- a.** Pinned
- b.** Default
- c.** Static
- d.** Dynamic

5. You are using the standard S3 storage class. In how many AZs is a data file stored by default?

- a.** Two
- b.** Greater than or equal to two
- c.** Greater than or equal to three
- d.** One

6. What security service should you consider HA designs for (at the very least) and might incorporate users or roles in AWS?

- a.** IAM
- b.** Application FW

- c. Network ACLs
 - d. Security Groups
7. Which statement regarding CloudWatch and HA is not true?
- a. Metrics are replicated across regions.
 - b. CloudWatch is done on a Region by Region basis.
 - c. CloudWatch is often a key ingredient with Elastic Load Balancing.
 - d. CloudWatch is often a key ingredient with Auto Scaling.

FOUNDATION TOPICS

HIGH AVAILABILITY COMPUTE

Before delving into the details of *high availability* (HA) in AWS, let's take a moment to discuss what this means. It's also important to point out that you should also understand fault tolerance (FT), which is a subset of HA.

High availability is a goal in solution design that focuses on automatic failover and recoverability from an issue in the solution. You typically measure HA using two metrics:

- ■ **RTO (Recovery Time Objective):** This metric seeks to define the amount of time in which the recovery must be completed; for example, an RTO of 4 hours indicates that your solution must be restored to full functionality in 4 hours or less.
- ■ **RPO (Recovery Point Objective):** This metric seeks to define the point to which data must be restored; this helps define how much data (for example) the organization is willing to lose; perhaps an organization is fine with losing 1 hour's worth of email; in this case, the RPO could be stated as

1 hour.

As previously stated, fault tolerance is a subset of high availability solutions. FT seeks to create a zero downtime solution and zero performance degradation due to a component failure. With a completely effective FT solution, the RTO and RPO values would both be 0.

The majority of AWS services are inherently fault tolerant. Non-VPC services (abstracted services) are almost all fault tolerant. For example, if you are building a solution using EC2, you automatically need to build out at least two EC2 instances across AZs to accomplish what most of the AWS services do natively. This is a big architectural consideration when comparing costs of AWS services. You need to consider the comparison of a fully redundant solution compared to the AWS service—for example, using S3 to host media instead of putting the media on EC2/EBS solutions and replicating. As another example, you might use Lambda to run cron processes versus an EC2 instance design running automation tasks. The EC2 instance would need to be set up in an HA/FT design to ensure automation does not fail.

If you are designing an HA solution for your EC2-based architecture, a simple and effective method for improving upon your high availability is to leverage the global infrastructure of AWS. You can do this by placing key resources in different regions and/or Availability Zones.

By provisioning EC2 instances in multiple geographic regions, you can launch Amazon EC2 instances in these regions, so your instances are closer to your customers. For example, you might want to launch instances in Europe to be closer to your

European customers or to help meet your legal requirements. But note that when you replicate resources in different regions, you are also improving your HA compute. If you have an instance in one Region fail, you can have the instance of another Region service the requests, even though latency might be higher.

Remember, each Region includes Availability Zones. Availability Zones are distinct locations that are engineered to be insulated from failures in other zones. They provide low latency network connectivity to other Availability Zones in the same region. By launching EC2 instances in separate Availability Zones, you can protect your applications from any failures that might affect an entire Availability Zone. To help achieve high availability, design your compute deployment to span two or more Availability Zones.

Elastic IP addresses are public IP addresses that can be programmatically mapped between EC2 instances within a Region. They are associated with the AWS account and not with a specific instance or lifetime of an instance. Elastic IP addresses can be used to work around host or Availability Zone failures by quickly remapping the address to another running instance or a replacement instance that was just started.

Lab: Provisioning EC2 Instances in Different Availability Zones



In this lab, you provision two identical EC2 instances but place each instance in a separate Availability Zone in a single

Region.

Step 1. In the AWS Management Console, select the **Region** drop-down menu and choose **South America (Sao Paulo)**.

Step 2. From the AWS Management Console, search for **EC2** and select the link.

Step 3. Click **Launch Instance**.

Step 4. Click the **Select** button for the Amazon Linux 2 AMI image.

Step 5. Click **Next: Configure Instance Details**.

Step 6. Click the **Subnet** drop-down menu and click the option for the **sa-east-1c** Availability Zone. Figure 5-1 demonstrates this configuration.

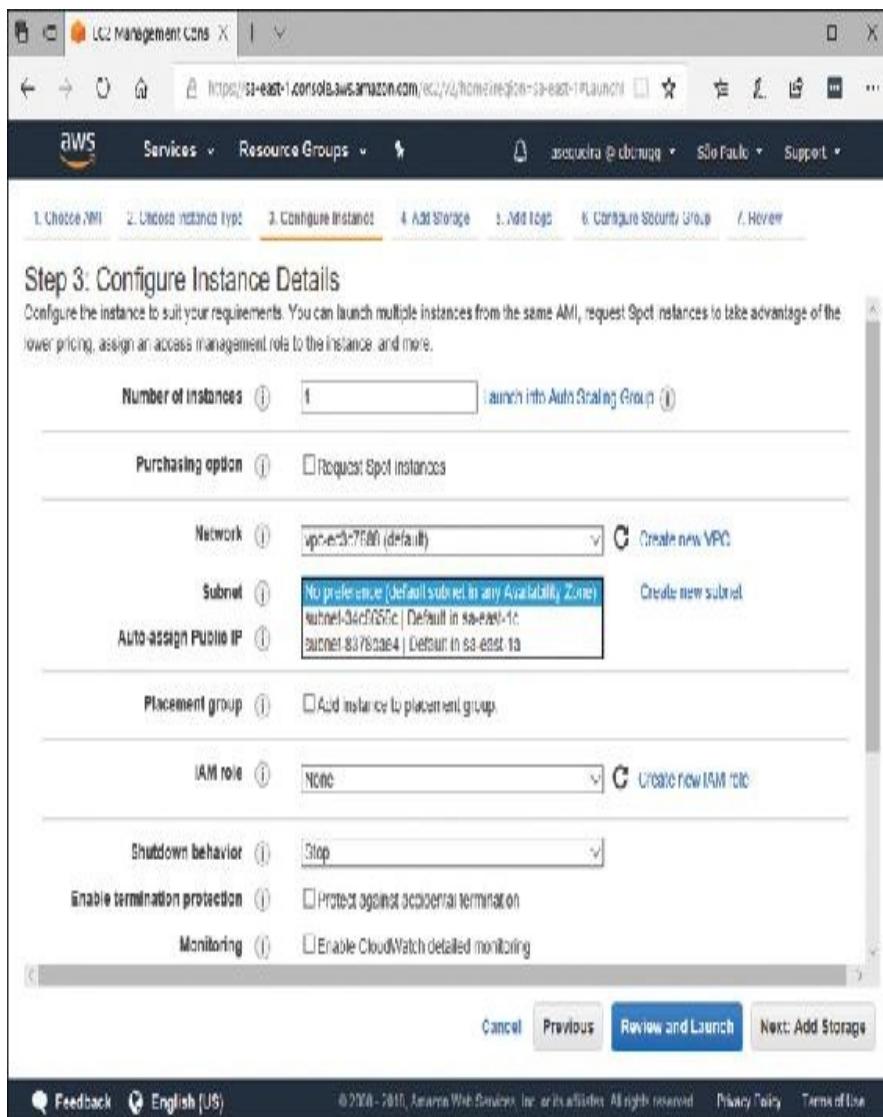


Figure 5-1 Placing an EC2 Instance in a Specific AZ

Step 7. Click **Next: Add Storage**.

Step 8. Click **Next: Add Tags**.

Step 9. Click **Next: Configure Security Group**.

Step 10. Click **Select an Existing Security Group** and choose the **Default** security group.

Step 11. Click **Review and Launch**.

Step 12. Click **Launch**.

Step 13. Click **Choose an Existing Key Pair** and select a key pair. Check the box that you acknowledge this key pair choice and choose **Launch Instances**.

Step 14. Choose **View Instances** to examine your running instance.

Step 15. Create another instance in another Availability Zone. To do so, click **Launch Instance**.

Step 16. Click the **Select** button for the Amazon Linux 2 AMI image.

Step 17. Click **Next: Configure Instance Details**.

Step 18. Click the **Subnet** drop-down menu and click the option for the **sa-east-1a** Availability Zone.

Step 19. Click **Next: Add Storage**.

Step 20. Click **Next: Add Tags**.

Step 21. Click **Next: Configure Security Group**.

Step 22. Click **Select an Existing Security Group** and choose the **Default** security group.

Step 23. Click **Review and Launch**.

Step 24. Click **Launch**.

Step 25. Click **Choose an Existing Key Pair** and select a key pair. Check the box that you acknowledge this key pair choice and choose **Launch Instances**.

Step 26. Choose **View Instances** to examine your running instances, as shown in Figure 5-2.

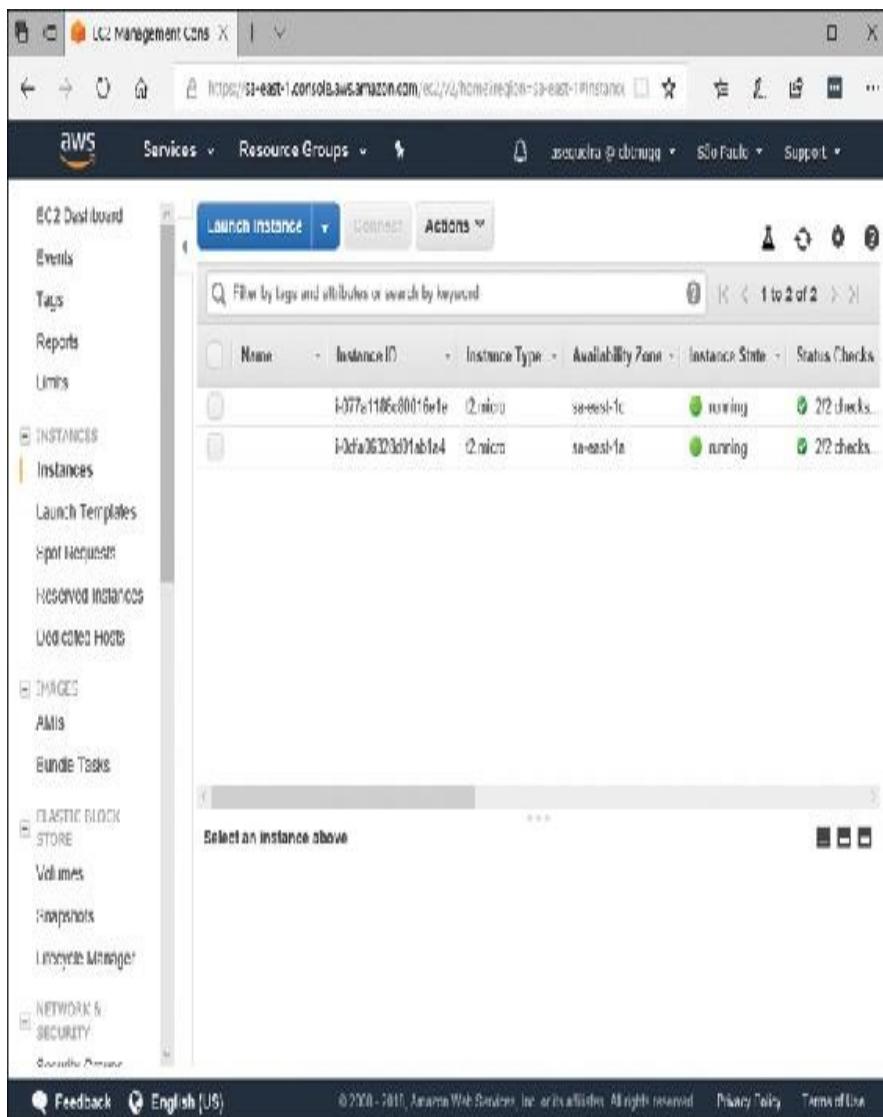


Figure 5-2 Viewing Your Running EC2 Instances

Lab Cleanup

Step 1. In the Instances area of EC2 (shown in Figure 5-2), click the check boxes for each instance to select them.

Step 2. Click the **Actions** drop-down menu and choose **Instance State** and then **Terminate**.

Step 3. Click **Yes, Terminate**.

HIGH AVAILABILITY APPLICATION SERVICES

Suppose that you start out running your application or website on a single EC2 instance, and over time, traffic increases to the point that you require more than one instance to meet the demand. You can launch multiple EC2 instances from your AMI and then use Elastic Load Balancing to distribute incoming traffic for your application across these EC2 instances. This increases the availability of your application. Placing your instances in multiple Availability Zones as done in the previous section also improves the fault tolerance in your application. If one Availability Zone experiences an outage, traffic is routed to the other Availability Zone.

You can use EC2 Auto Scaling to maintain a minimum number of running instances for your application at all times. EC2 Auto Scaling can detect when your instance or application is unhealthy and replace it automatically to maintain the availability of your application. You can also use EC2 Auto Scaling to scale your EC2 capacity out or in automatically based on demand, using criteria that you specify.

Chapter 9, “Designing for Elasticity,” covers Elastic Load Balancing and Auto Scaling in detail.

HIGH AVAILABILITY DATABASE SERVICES

Relational Database Services (RDS) provides high availability and failover support for DB instances using Multi-AZ deployments. RDS uses several different technologies to

provide failover support. Multi-AZ deployments for Oracle, PostgreSQL, MySQL, and MariaDB instances use Amazon's failover technology. SQL Server DB instances use SQL Server Mirroring.

In a Multi-AZ deployment, RDS automatically provisions and maintains a synchronous standby replica in a different Availability Zone. The primary DB instance is synchronously replicated across Availability Zones to a standby replica to provide data redundancy, eliminate I/O freezes, and minimize latency spikes during system backups. Running a DB instance with high availability can enhance availability during planned system maintenance and help protect your databases against DB instance failure and Availability Zone disruption.

Note

The high availability feature is not a scaling solution for read-only scenarios; you cannot use a standby replica to serve read traffic. To service read-only traffic, you should use a Read Replica.

Using the RDS console, you can create a Multi-AZ deployment by simply specifying Multi-AZ when creating a DB instance. You can also use the console to convert existing DB instances to Multi-AZ deployments by modifying the DB instance and specifying the Multi-AZ option. The RDS console shows the Availability Zone of the standby replica, called the secondary AZ.

You can specify a Multi-AZ deployment using the CLI as well. Use the AWS CLI **describedb-instances** command or the Amazon RDS API **DescribeDBInstances** action to show the Availability Zone of the standby replica.

DB instances using Multi-AZ deployments may have increased write and commit latency compared to a Single-AZ deployment, due to the synchronous data replication that occurs. You may have a change in latency if your deployment fails over to the standby replica, although AWS is engineered with low-latency network connectivity between Availability Zones. For production workloads, Amazon recommends that you use Provisioned IOPS and DB instance classes (m1.large and larger) that are optimized for Provisioned IOPS for fast, consistent performance.

If you have a DB instance in a Single-AZ deployment and you modify it to be a Multi-AZ deployment (for engines other than SQL Server or Aurora), RDS takes several steps:

Step 1. RDS takes a snapshot of the primary DB instance from your deployment and then restores the snapshot into another Availability Zone.



Step 2. RDS then sets up synchronous replication between your primary DB instance and the new instance. This action avoids downtime when you convert from Single-AZ to Multi-AZ, but you can experience a significant performance impact when first converting to Multi-AZ. This impact is more noticeable for large and write-intensive DB instances.

Step 3. Once the modification is complete, RDS triggers an event (RDS-EVENT-0025) that indicates the process is complete.

In the event of a planned or unplanned outage of your DB instance, RDS automatically switches to a standby replica in another Availability Zone if you have enabled Multi-AZ. The time it takes for the failover to complete depends on the database activity and other conditions at the time the primary DB instance became unavailable. Failover times are typically 60–120 seconds. However, large transactions or a lengthy recovery process can increase failover time. When the failover is complete, it can take additional time for the RDS console UI to reflect the new Availability Zone.

The failover mechanism automatically changes the DNS record of the DB instance to point to the standby DB instance. As a result, you need to re-establish any existing connections to your DB instance. Because of the way the Java DNS caching mechanism works, you may need to reconfigure your JVM environment.

RDS handles failovers automatically so you can resume database operations as quickly as possible without administrative intervention.



The primary DB instance switches over automatically to the standby replica if any of the following conditions occur:

- ■ An Availability Zone has an outage.
- ■ The primary DB instance fails.
- ■ The DB instance's server type is changed.
- ■ The operating system of the DB instance is undergoing software patching.
- ■ A manual failover of the DB instance was initiated using Reboot with failover.

There are several ways to determine if your Multi-AZ DB instance has failed over:

- ■ You can set up DB event subscriptions to notify you via email or SMS that a failover has been initiated.
- ■ You can view your DB events by using the RDS console or API actions.
- ■ You can view the current state of your Multi-AZ deployment by using the RDS console and API actions.

Note

Not all regions support all DB engines in a Multi-AZ deployment. For example, some AZs support Multi-AZ for all engines except MS SQL Server. This can be problematic if customers decide they want Multi-AZ after development and realize they cannot turn on this feature. It is critical for customers who require (or might require in the future) Multi-AZ that they not only confirm that RDS supports Multi-AZ in the region in question but also that the specific engine supports it in that Region.

HIGH AVAILABILITY NETWORKING SERVICES

A frequent design goal is to provide high availability connectivity between your data center and your VPC. When possible, AWS recommends connecting from multiple data centers to add physical-location redundancy, in addition to hardware redundancy.

Consider the following best practices:

- ■ Leverage multiple dynamically routed, rather than statically routed, connections to AWS. This allows remote connections to fail over automatically between redundant connections. Dynamic routing also enables remote connections to automatically leverage available preferred routes, if applicable, to the on-premises network.
- ■ Avoid relying on a single on-premises device, even if you were planning to use multiple interfaces for availability. Highly available connections require redundant hardware, even when connecting from the same physical

location.

- ■ When selecting AWS Direct Connect network service providers, consider a dual-vendor approach, if financially feasible, to ensure private-network diversity.
- ■ Native Direct Connect can be established without a provider. A native Direct Connect deployment has the organization place its own hardware in an AWS partner data center. Customers would rack their own equipment. Separate Direct Connect equipment within the same partner data center could be requested, or separate data center connection points could also be requested. If a third-party provider is chosen for Direct Connect, at the very least ensure the partner has terminations in different data centers.
- ■ Provision sufficient network capacity to ensure that the failure of one network connection does not overwhelm and degrade redundant connections.

Many AWS customers choose to implement VPN connections to set up remote connectivity to a VPC. To enable redundancy, each AWS Virtual Private Gateway has two VPN endpoints with capabilities for static and dynamic routing. Although statically routed VPN connections from a single customer gateway are sufficient for establishing remote connectivity to a VPC, this is not a highly available configuration. The best practice for making VPN connections highly available is to use redundant customer gateways and dynamic routing for automatic failover between AWS and customer VPN endpoints.

You might also establish private connectivity between AWS and your data center, office, or colocation environment with AWS Direct Connect to reduce network costs, increase bandwidth throughput, or provide a more consistent network experience than Internet-based connections. Because each dedicated, physical connection is in one AWS Direct Connect location, multiple dynamically routed AWS Direct Connect connections are necessary to achieve high availability.

Architectures with the highest levels of availability will leverage different AWS Direct Connect partner networks to ensure network-provider redundancy.

You might prefer the benefits of one or more AWS Direct Connect connections for their primary connectivity to AWS, coupled with a lower-cost backup connection. To achieve this objective, they can establish AWS Direct Connect connections with a VPN backup.

HIGH AVAILABILITY STORAGE SERVICES

Remember, AWS provides many HA mechanisms by default with your storage solutions. For example, the S3 service (with most of the storage classes) seeks to automatically achieve HA by placing multiple copies of your data in different Availability Zones.

Table 5-2 provides an overview of the different storage classes and the HA levels and AZs used with each.

Table 5-2 S3 High Availability

Key Topic				
S3 Standard	S3 Standard-IA	S3 One Zone-IA	S3 Glacier	
Availability	99.99%	99.9%	99.5%	NA

Availability Zones	Greater than or equal to three	Greater than or equal to three	One	Greater than or equal to three

HIGH AVAILABILITY SECURITY SERVICES

While you might strive for HA in areas like compute and applications, you might not immediately think of your security mechanisms with AWS. Remember to consider this area as well, especially when it comes to the critical component of IAM and your security. Administrators can make mistakes, such as incorrectly modifying an IAM policy or inadvertently disabling an access key. If you design AWS applications, you should apply the same high availability concepts to your use of IAM so that the impact of these events can be minimized or avoided completely.

In general, you can partition the authentication elements (such as an IAM user or role) and authorization elements (such as IAM policies) for a single application into multiple sets. Therefore, if a change is made to the authentication or authorization configuration for a single set, any negative impact would be confined to that set rather than affecting the entire application.

For example, you might take the application instances running behind a load balancer and designate a small set that you can use for production validation. That way, to make sure a change you have tested in preproduction does not affect the whole

cluster, you can deploy a new IAM policy to just this small set. After you have validated that the change has not introduced any regressions, you can deploy the change to the remaining instances with confidence.

HIGH AVAILABILITY MONITORING SERVICES

As you have most likely already considered, HA solutions in AWS require extensive use of monitoring capabilities like CloudWatch. Because this is the case, one could argue that all your HA designs hinge upon the high availability capabilities of monitoring itself. Fortunately, AWS cloud computing resources are housed in highly available data center facilities. As you know, to provide additional scalability and reliability, each data center facility is located in a specific geographical area known as a Region. Each Region is designed to be completely isolated from the other Regions, to achieve the greatest possible failure isolation and stability. CloudWatch does not aggregate data across regions. Therefore, metrics are completely separate between regions, and you should consider this in your overall design.

EXAM PREPARATION TASKS

As mentioned in the section “How to Use This Book” in the Introduction, you have a couple of choices for exam preparation: the exercises here, Chapter 16, “Final Preparation,” and the exam simulation questions in the Pearson Test Prep practice test software.

REVIEW ALL KEY TOPICS

Review the most important topics in this chapter, noted with the Key Topics icon in the margin of the page. Table 5-3 lists a reference of these key topics and the page numbers on which each is found.

Table 5-3 Key Topics for Chapter 5

Key Topic		
Key Topic Element	Description	Page Number
Lab	Provisioning EC2 Instances in Different Availability Zones	85
Steps	HA in RDS	89
List	Triggers for primary DB switchover	90
Table 5-2	S3 High Availability	92

COMPLETE TABLES AND LISTS FROM MEMORY

Print a copy of Appendix C, “Memory Tables” (found on the

book website), or at least the section for this chapter, and complete the tables and lists from memory. Appendix D, “Memory Tables Answer Key,” also on the website, includes completed tables and lists to check your work.

DEFINE KEY TERMS

Define the following key terms from this chapter and check your answers in the glossary:

HA

Multi-AZ

FT

RPO

RTO

Q&A

The answers to these questions appear in Appendix A. For more practice with exam format questions, use the Pearson Test Prep practice test software.

1. What are two services that are often critical for high availability and that ensure resource creation and load sharing based on demand?
2. What is a redundant copy of a database called and is a reference to replication?
3. What are different geographically separate data center clusters in a Region called in AWS?

Part II

Domain 2: Define Performant Architectures

Chapter 6. Choosing Performant Storage

This chapter covers the following subjects:

- **■ Performant S3 Services:** This section is critical thanks to the widespread use of S3. Here, you learn key tips for keeping S3 running with the performance you require.
- **■ Performant EBS Services:** EBS performance is a complex topic. This section provides valuable architectural concepts.
- **■ Performant EFS Services:** EFS permits excellent performance if you know the correct components for doing so. This section ensures you can tune EFS as needed.
- **■ Performant Glacier Services:** This section provides key guidance regarding your Vaults within Glacier.

More and more data is consistently stored in the cloud. This data continues to vary in content and requirements. It is critical that you know how to architect the appropriate storage for the appropriate scenarios. And when you do so, you also must ensure the storage performs as expected and required. This chapter delves deep into the areas of storage performance.

“DO I KNOW THIS ALREADY?” QUIZ

The “Do I Know This Already?” quiz allows you to assess whether you should read the entire chapter. Table 6-1 lists the major headings in this chapter and the “Do I Know This Already?” quiz questions covering the material in those headings so you can assess your knowledge of these specific areas. The answers to the “Do I Know This Already?” quiz appear in [Appendix A](#).

Table 6-1 “Do I Know This Already?” Foundation Topics
Section-to-Question Mapping

Foundation Topics Section	Question
Performant S3 Services	1
Performant EBS Services	2
Performant EFS Services	3
Performant Glacier Services	4

Caution

The goal of self-assessment is to gauge your mastery of the topics in this chapter. If you do not know the answer to a question or are only partially sure of the answer, you should mark that question as wrong for purposes of the self-assessment. Giving yourself credit for an answer you correctly guess skews your self-assessment results and might provide you with a false sense of security.

1. It has been reported to you that your S3 performance is suffering. The bucket is filled automatically with log files from one of your AWS architected solutions. What might be the issue?

- a. The device is creating key names that all begin the same.
 - b. The device is requiring NFS usage.
 - c. S3 is not an ideal repository for log files.
 - d. The device needs to be moved to the cloud.
- 2. With what type of storage does Amazon recommend you configure the read ahead setting to 1 MB?
 - a. EBS HDD Volumes
 - b. S3 Buckets
 - c. EBS SSD-backed Volumes
 - d. EFS Systems
- 3. Which version of NFS provides the best performance?
 - a. Version 3.0
 - b. Version 4.1
 - c. Version 2.5
 - d. Version 8
- 4. How long do expedited retrievals take in Glacier?
 - a. 1 to 5 minutes
 - b. 1 to 3 hours
 - c. 1 to 2 days
 - d. 5 to 12 hours

FOUNDATION TOPICS

PERFORMANT S3 SERVICES

You should consider the following guidelines when it comes to architecting performant S3 solutions:

- ■ S3 automatically scales to high request rates. For example, your application can achieve at least 3,500 PUT/POST/DELETE and 5,500 GET requests per second per prefix in a bucket. There are no limits to the number of prefixes in a bucket. It is simple to increase your read or write performance exponentially. For example, if you create 10 prefixes in an S3 bucket to parallelize reads, you could scale your read performance to 55,000 read requests per second.
- ■ If your S3 workload uses server-side encryption with AWS Key Management Service (KMS), you should reference the “AWS KMS Limits” section of the *AWS Key Management Service Developer Guide* for information about the request rates supported for your use case.
- ■ If your workload is mainly sending GET requests, you should consider using CloudFront in conjunction with S3 for performance optimization. With CloudFront, you can distribute content to your users with low latency and a high data transfer rate. You also send fewer direct requests to S3. This can reduce storage costs.
- ■ TCP window scaling allows you to improve network throughput performance between your operating system and application layer and S3 by supporting window sizes larger than 64 KB. At the start of the TCP session, a client advertises its supported receive window WSCALE factor, and S3 responds with its supported receive window WSCALE factor for the upstream direction.
- ■ TCP selective acknowledgment is designed to improve recovery time after a large number of packet losses. TCP selective acknowledgment is supported by most newer operating systems but might have to be enabled.

It is also critical to keep this in mind for very high workload environments: S3 maintains an index of object key names in each region. Object keys are stored in UTF-8 binary ordering across multiple partitions in the index. The key name determines which partition the key is stored in. Using a sequential prefix, such as a timestamp or a number sequence, increases the likelihood that S3 will target a specific partition

for a large number of your keys. This can overwhelm the I/O capacity of the partition.

Key Topic

The solution to this problem is simple. If your workload is a mix of request types, introduce some randomness to key names by adding a hash string as a prefix to the key name. When you introduce randomness to your key names, the I/O load is distributed across multiple index partitions. The following example shows key names with a four-character hexadecimal hash added as a prefix:

```
myexampleawsbucket/232a-2019-14-03-15-00-  
00/cust123/photo1.jpg  
myexampleawsbucket/7b54-2019-14-03-15-00-  
00/cust123/photo2.jpg  
myexampleawsbucket/921c-2019-14-03-15-00-  
00/cust345/photo3.jpg  
myexampleawsbucket/ba65-2019-14-03-15-00-  
00/cust345/photo4.jpg  
myexampleawsbucket/8761-2019-14-03-15-00-  
00/cust345/photo5.jpg
```

Without the four-character hash prefix, S3 might distribute all of this load to one or two index partitions because the name of each object begins the same and all objects in the index are stored in alphanumeric order. The four-character hash prefix ensures that the load is spread across multiple index partitions.

If your workload is sending mainly GET requests, you can add randomness to key names. You can also integrate CloudFront with S3 to distribute content to your users with low latency and a high data transfer rate.

Note

Amazon has offered increased performance for S3 across all regions and has applied this to all S3 implementations with no intervention required from you. Amazon made this announcement: "This S3 request rate performance increase removes any previous guidance to randomize object prefixes to achieve faster performance. That means you can now use logical or sequential naming patterns in S3 object naming without any performance implications." Amazon has indicated it will be modifying its certification exams to reflect this. We decided to keep the "legacy" information in this text regarding sequential naming patterns in case you still encounter such questions on your test date.

You can also optimize performance when uploading data to an S3 bucket or transferring data between S3 buckets. The S3 console provides a simple interface for uploading and copying relatively small amounts of data to S3 buckets. If you need to work with large volumes of data, you can improve performance by using other methods.

If the S3 buckets are in the same region, you can use the AWS CLI to simultaneously run multiple instances of the **S3 cp** (copy), **mv** (move), or **sync** (synchronize) commands with the **--exclude** filter to increase performance through multithreading.

The AWS CLI can also utilize stdout and stdin, allowing you to "stream" data through it without needing to have the necessary storage on the local EC2 instance (avoiding writing to disk). For example, bucketa is copied to bucketb via an EC2 instance with the CLI installed. bucketa has an object that is 5 TB that you want to copy to bucketb. You don't need 5 TB on the local EC2 instance to do this. You use the - option to stream to stdout and then pipe that into another command where the source is stdin. The CLI would use the following command:

```
aws s3 cp s3://bucketa/image.png - | aws s3 cp -  
s3://bucketb/image.png
```

Do you have a large amount to upload or download to your S3 implementation that will take more than a week to complete given your bandwidth limitations? You should also consider using Snowball for these S3 uploads or downloads. Snowball is a petabyte-scale data transport solution that uses secure appliances to transfer large amounts of data into and out of AWS.

PERFORMANT EBS SERVICES

With instances in EC2, several factors, including I/O characteristics and the configuration of your instances and volumes, can affect the performance of EBS. In some cases you might need to do some tuning to achieve peak performance.

Wisely, Amazon recommends that you tune performance with information from your actual workload, in addition to benchmarking, to determine your optimal configuration. After you learn the basics of working with EBS volumes, it is a good idea to look at the I/O performance you require and at your options for increasing EBS performance to meet those requirements.

Consider the following guidelines to ensure you architect the best possible performance in your EBS services:

- On instances without support for EBS-optimized throughput, network traffic can contend with traffic between your instance and your EBS volumes; on EBS-optimized instances, the two types of traffic are kept separate.

- When you measure the performance of your EBS volumes, it is important to understand the units of measure involved and how performance is calculated. For example, IOPS are a unit of measure representing input/output operations per second. The operations are measured in KB, and the underlying drive technology determines the maximum amount of data that a volume type counts as a single I/O. I/O size is capped at 256 KB for SSD volumes and 1,024 KB for HDD volumes because SSD volumes handle small or random I/O much more efficiently than HDD volumes. When small I/O operations are physically contiguous, EBS attempts to merge them into a single I/O up to the maximum size.
- There is a relationship between the maximum performance of your EBS volumes, the size and number of I/O operations, and the time it takes for each action to complete. Each of these factors affects the others, and different applications are more sensitive to one factor or another.
- There is a significant increase in latency when you first access each block of data on a new EBS volume that was restored from a snapshot. You can avoid this performance hit by accessing each block prior to putting the volume into production. This process is called initialization (pre-warming). Some tools that exist will perform reads of all the blocks for you in order to achieve this.
- When you create a snapshot of a Throughput Optimized HDD (st1) or Cold HDD (sc1) volume, performance may drop as far as the volume's baseline value while the snapshot is in progress. This behavior is specific to these volume types.
- Other factors that can limit performance include driving more throughput than the instance can support, the performance penalty encountered while initializing volumes restored from a snapshot, and excessive amounts of small, random I/O on the volume.
- Your performance can also be affected if your application is not sending enough I/O requests. You can monitor this by looking at your volume's queue length and I/O size. The queue length is the number of pending I/O requests from your application to your volume. For maximum consistency, HDD-backed volumes must maintain a queue length (rounded to the nearest whole number) of 4 or more when performing 1 MB sequential I/O.
- Some workloads are read-heavy and access the block device through the operating system page cache (for example, from a file system). In this case, if you want to achieve the maximum throughput, Amazon recommends that

you configure the read-ahead setting to 1 MB. This is a per-block-device setting that should be applied only to your HDD volumes.

- ■ Some instance types can drive more I/O throughput than what you can provide for a single EBS volume. You can join multiple gp2, io1, st1, or sc1 volumes together in a RAID 0 configuration to use the available bandwidth for these instances.
- ■ When you plan and configure EBS volumes for your application, it is important to consider the configuration of the instances that you will attach the volumes to. To get the most performance from your EBS volumes, you should attach them to an instance with enough bandwidth to support your volumes, such as an EBS-optimized instance or an instance with 10 Gigabit network connectivity. This is especially important when you stripe multiple volumes together in a RAID configuration.
- ■ Launching an instance that is EBS-optimized provides you with a dedicated connection between your EC2 instance and your EBS volume. However, it is still possible to provision EBS volumes that exceed the available bandwidth for certain instance types, especially when multiple volumes are striped in a RAID configuration. Be sure to choose an EBS-optimized instance that provides more dedicated EBS throughput than your application needs; otherwise, the EBS to EC2 connection becomes a performance bottleneck.
- ■ On a given volume configuration, certain I/O characteristics drive the performance behavior for your EBS volumes. SSD-backed volumes—General Purpose SSD (gp2) and Provisioned IOPS SSD (io1)—deliver consistent performance whether an I/O operation is random or sequential. HDD-backed volumes—Throughput Optimized HDD (st1) and Cold HDD (sc1)—deliver optimal performance only when I/O operations are large and sequential.
- ■ The volume queue length is the number of pending I/O requests for a device. Latency is the true end-to-end client time of an I/O operation—in other words, the time elapsed between sending an I/O to EBS and receiving an acknowledgment from EBS that the I/O read or write is complete. Queue length must be correctly calibrated with I/O size and latency to avoid creating bottlenecks either on the guest operating system or on the network link to EBS. Optimal queue length varies for each workload, depending on your particular application's sensitivity to IOPS and latency.
- ■ For SSD-backed volumes, if your I/O size is very large, you may

experience a smaller number of IOPS than you provisioned because you are hitting the throughput limit of the volume.

PERFORMANT EFS SERVICES

This section provides an overview of EFS performance, discusses the available performance modes and throughput modes, and outlines some useful performance tips.

EFS file systems are distributed across an unconstrained number of storage servers, enabling file systems to grow elastically to petabyte scale and allowing massively parallel access from EC2 instances to your data. The distributed design of EFS avoids the bottlenecks and constraints inherent to traditional file servers.

This distributed data storage design means that multithreaded applications and applications that concurrently access data from multiple EC2 instances can drive substantial levels of aggregate throughput and IOPS. Big data and analytics workloads, media processing workflows, content management, and web serving are examples of these applications.

In addition, EFS data is distributed across multiple Availability Zones, providing a high level of durability and availability. Table 6-2 compares high-level performance and storage characteristics for Amazon's file and block cloud storage services.

Table 6-2 EFS vs. EBS



EFS

EBS Provisioned IOPS

Per-operation latency	Low, consistent latency	Lowest consistent latency
Throughput scale	10+ GB per second	Up to 2 GB per second
Availability and durability	Redundant across multiple AZs	Redundant within a single AZ
Access	Thousands	Single instance

This distributed architecture results in a small latency overhead for each file operation. Due to this per-operation latency, overall throughput increases as the average I/O size increases because the overhead is amortized over a larger amount of data. EFS supports highly parallelized workloads (for example, using concurrent operations from multiple threads and multiple EC2 instances), which enables high levels of aggregate throughput and operations per second.

To support a wide variety of cloud storage workloads, EFS offers two performance modes. You select a file system's performance mode when you create it, as shown in Figure 6-1.

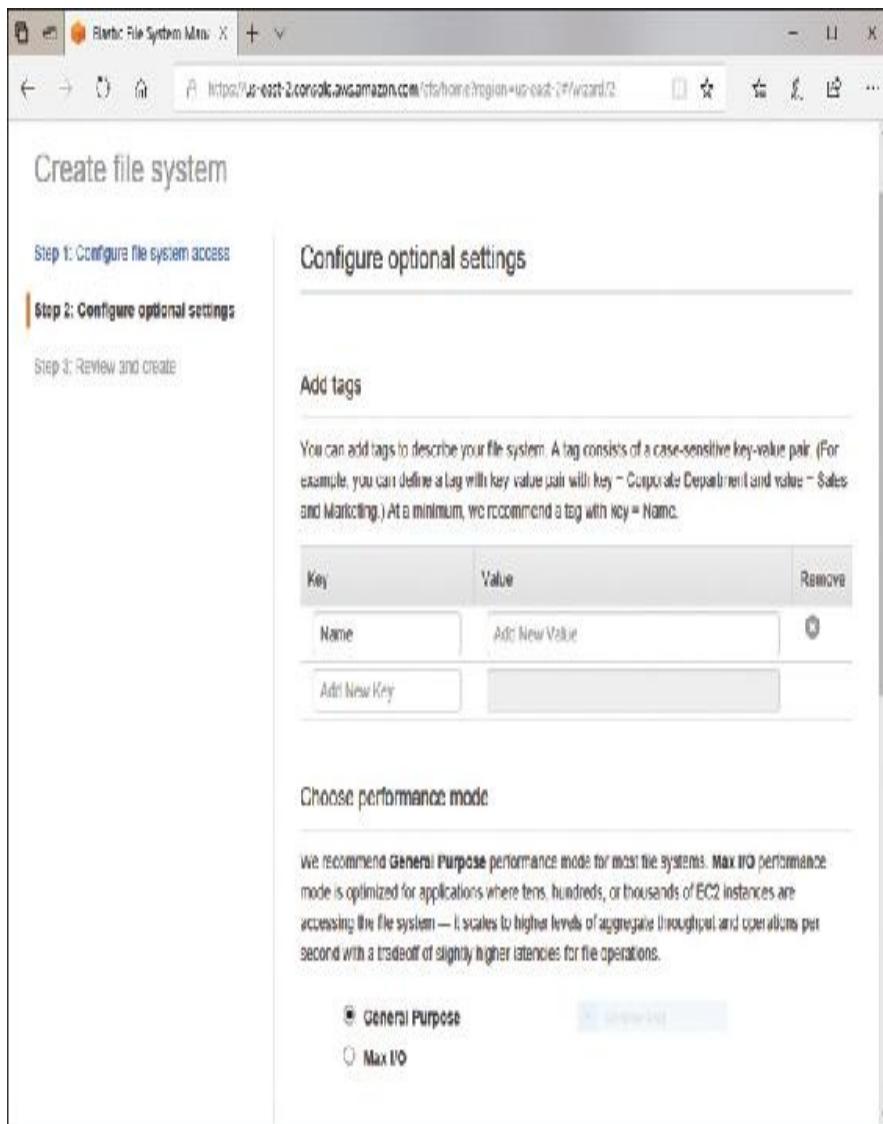


Figure 6-1 Choose the EFS Performance Mode

Amazon recommends General Purpose performance mode for the majority of your EFS file systems. General Purpose is ideal for latency-sensitive use cases, like web-serving environments, content management systems, home directories, and general file serving. If you do not choose a performance mode when you create your file system, EFS selects the General Purpose mode for you by default.

File systems in the Max I/O mode can scale to higher levels of

aggregate throughput and operations per second with a trade-off of slightly higher latencies for file operations. Highly parallelized applications and workloads, such as big data analysis, media processing, and genomics analysis, can benefit from this mode.

There are two throughput modes to choose from for your file system: Bursting Throughput and Provisioned Throughput.

With Bursting Throughput mode, throughput on EFS scales as your file system grows. With Provisioned Throughput mode, you can instantly provision the throughput of your file system (in MB/s) independent of the amount of data stored.

With Bursting Throughput mode, throughput on EFS scales as a file system grows. File-based workloads are typically spiky, driving high levels of throughput for short periods of time, and low levels of throughput the rest of the time. To accommodate this, EFS is designed to burst to high throughput levels for periods of time.

All file systems, regardless of size, can burst to 100 MB/s of throughput. Those over 1 TB large can burst to 100 MB/s per TB of data stored in the file system. For example, a 10 TB file system can burst to 1,000 MB/s of throughput ($10\text{ TB} \times 100\text{ MB/s/TB}$). The portion of time a file system can burst is determined by its size. The bursting model is designed so that typical file system workloads can burst virtually any time they need to.

EFS uses a credit system to determine when file systems can burst. Each file system earns credits over time at a baseline rate that is determined by the size of the file system and uses credits whenever it reads or writes data. The baseline rate is 50

MB/s per TB of storage (equivalently, 50 KB/s per GB of storage).

Accumulated burst credits give the file system the ability to drive throughput above its baseline rate. A file system can drive throughput continuously at its baseline rate, and whenever it's inactive or driving throughput below its baseline rate, the file system accumulates burst credits.

When a file system has a positive burst credit balance, it can burst. You can see the burst credit balance for a file system by viewing the `BurstCreditBalance` CloudWatch metric for EFS.

The bursting capability (both in terms of length of time and burst rate) of a file system is directly related to its size. Larger file systems can burst at larger rates for longer periods of time. In some cases, your application might need to burst more (that is, you might find that your file system is running out of burst credits). In these cases, you should increase the size of your file system or switch to Provisioned Throughput mode.

Use your historical throughput patterns to calculate the file system size you need to sustain your desired level of activity.

Provisioned Throughput mode is available for applications with high throughput to storage (MB/s per TB) ratios or with requirements greater than those allowed by the Bursting Throughput mode.

If your file system is in the Provisioned Throughput mode, you can increase the Provisioned Throughput of your file system as often as you want. You can decrease your file system throughput in Provisioned Throughput mode as long as it has been more than 24 hours since the last decrease. Additionally,

you can change between Provisioned Throughput mode and the default Bursting Throughput mode as long as it's been more than 24 hours since the last throughput mode change.

If your file system in the Provisioned Throughput mode grows in size after initial configuration, your file system might have a higher baseline rate in Bursting Throughput mode. In that case, your file system throughput is set to the amount that it's entitled to in the Bursting Throughput mode. You do not incur an additional charge for the throughput beyond the bursting storage cost. You also can burst for additional throughput to values defined by Bursting Throughput mode.

By default, Amazon recommends that you run your application in the Bursting Throughput mode. If you experience performance issues, check the `BurstCreditBalance` CloudWatch metric. If the value of the `BurstCreditBalance` metric is either zero or steadily decreasing, Provisioned Throughput is right for your application.

The Bursting Throughput model for the EFS file systems remains the same whether accessed from your on-premises servers or your EC2 instances. However, when you're accessing EFS file data from your on-premises servers, the maximum throughput is also constrained by the bandwidth of the AWS Direct Connect connection.

Because of the propagation delay tied to data traveling over long distances, the network latency of an AWS Direct Connect connection between your on-premises data center and your VPC can be tens of milliseconds. If your file operations are serialized, the latency of the AWS Direct Connect connection directly impacts your read and write throughput. The volume

of data you can read or write during a period of time is bounded by the amount of time it takes for each read and write operation to complete. To maximize your throughput, parallelize your file operations so that multiple reads and writes are processed by EFS concurrently.

When using EFS, keep the following performance tips in mind:

- **Average I/O Size:** The distributed nature of EFS enables high levels of availability, durability, and scalability. This distributed architecture results in a small latency overhead for each file operation. Due to this per-operation latency, overall throughput increases as the average I/O size increases because the overhead is amortized over a larger amount of data.
- **Simultaneous Connections:** EFS file systems can be mounted on up to thousands of EC2 instances concurrently. If you can parallelize your application across more instances, you can drive higher throughput levels on your file system in aggregate across instances.
- **Request Model:** By enabling asynchronous writes to your file system, pending write operations are buffered on the EC2 instance before they are written to EFS asynchronously. Asynchronous writes typically have lower latencies. When performing asynchronous writes, the kernel uses additional memory for caching. A file system that has enabled synchronous writes, or one that opens files using an option that bypasses the cache (for example, O_DIRECT), issues synchronous requests to EFS. Every operation goes through a round trip between the client and EFS.
- **NFS Client Mount Settings:** Verify that you are using the recommended mount options as outlined in Mounting File Systems and in Additional Mounting Considerations. EFS supports the Network File System versions 4.0 and 4.1 (NFSv4) and NFSv4.0 protocols when mounting your file systems on EC2 instances. NFSv4.1 provides better performance.
- **EC2 Instances:** Applications that perform a large number of read and write operations likely need more memory or computing capacity than applications that do not. When launching your EC2 instances, choose instance types that have the amount of these resources that your application needs. The performance characteristics of EFS file systems do not depend on the use of EBS-optimized instances.

- **Encryption:** EFS supports two forms of encryption: encryption in transit and encryption at rest. This option is for encryption at rest. Choosing to enable either or both types of encryption for your file system has a minimal effect on I/O latency and throughput.

PERFORMANT GLACIER SERVICES

Architecting Glacier services for performance is much simpler because Amazon is managing the service heavily. With that said, keep these tips in mind:



- When uploading large archives (100MB or larger), you can use multipart upload to achieve higher throughput and reliability. Multipart uploads allow you to break your large archive into smaller chunks that are uploaded individually. After all the constituent parts are successfully uploaded, they are combined into a single archive.
- Standard retrievals allow you to access any of your archives within several hours. Standard retrievals typically complete within 3–5 hours.
- Bulk retrievals are Glacier's lowest-cost retrieval option, enabling you to retrieve large amounts, even petabytes, of data inexpensively in a day. Bulk retrievals typically complete within 5–12 hours.
- Expedited retrievals allow you to quickly access your data when occasional urgent requests for a subset of archives are required. For all but the largest archives (250 MB+), data accessed using expedited retrievals is typically made available within 1–5 minutes. There are two types of expedited retrievals: On-Demand and Provisioned.
- On-Demand requests are like EC2 On-Demand instances and are available the vast majority of the time.
- Provisioned Capacity guarantees that your retrieval capacity for expedited retrievals will be available when you need it. Each unit of capacity ensures that at least three expedited retrievals can be performed every 5 minutes and provides up to 150 MB/s of retrieval throughput.

EXAM PREPARATION TASKS

As mentioned in the section “How to Use This Book” in the Introduction, you have a couple of choices for exam preparation: the exercises here, Chapter 16, “Final Preparation,” and the exam simulation questions in the Pearson Test Prep practice test software.

REVIEW ALL KEY TOPICS

Review the most important topics in this chapter, noted with the Key Topics icon in the margin of the page. Table 6-3 lists a reference of these key topics and the page numbers on which each is found.

Table 6-3 Key Topics for Chapter 6

		
Key Topic Element	Description	Page Number
Concept	S3 and Key Names	99
Table 6-2	EFS vs. EBS	104
List	Tips for Glacier Performance	108

COMPLETE TABLES AND LISTS FROM

MEMORY

Print a copy of Appendix C, “Memory Tables” (found on the book website), or at least the section for this chapter, and complete the tables and lists from memory. Appendix D, “Memory Tables Answer Key,” also on the website, includes completed tables and lists to check your work.

DEFINE KEY TERMS

Define the following key terms from this chapter and check your answers in the glossary:

TCP Window Scaling

TCP Selective Acknowledgment

Initialization

Latency

EBS-optimized instance

Volume queue length

Bursting Throughput Mode

Provisioned Throughput Mode

Q&A

The answers to these questions appear in Appendix A. For more practice with exam format questions, use the Pearson Test Prep practice test software.

1. What is the best approach to solve performance issues that result from key names in S3?
2. For fast transfer between S3 buckets in the same region, what is the recommended configuration tool?

- 3.** On an EBS-optimized volume, how is contention solved?

- 4.** Standard retrievals from Glacier can take about how much time?

Chapter 7. Choosing Performant Databases

This chapter covers the following subjects:

- **Aurora:** Amazon built Aurora with performance in mind. This section ensures you use best practices to experience the best possible performance.
- **RDS:** Although RDS has many database engines to choose from, and each of these engines features its own performance best practices, this section covers what you should know in general regarding the best possible performance in RDS.
- **DynamoDB:** Because DynamoDB uses different methods for data management and storage than you might be accustomed to, this section ensures you can build high-performance DynamoDB environments.
- **ElastiCache:** This section covers caching strategies and other best practices for using ElastiCache in your architecture.
- **Redshift:** Because Redshift is unique compared to other SQL database systems, this section provides interesting best practices for the best possible performance.

Databases are often a key part of your AWS architecture. Fortunately, no matter what your database service of choice in AWS, you can use concrete and easily implemented best practices to squeeze the best performance out of these key constructs.

“DO I KNOW THIS ALREADY?” QUIZ

The “Do I Know This Already?” quiz allows you to assess whether you should read the entire chapter. Table 7-1 lists the major headings in this chapter and the “Do I Know This Already?” quiz questions covering the material in those headings so you can assess your knowledge of these specific

areas. The answers to the “Do I Know This Already?” quiz appear in [Appendix A](#).

Table 7-1 “Do I Know This Already?” Foundation Topics
Section-to-Question Mapping

Foundation Topics Section	Question
Aurora	1–2
RDS	3–4
DynamoDB	5–6
ElastiCache	7–8
Redshift	9–10

Caution

The goal of self-assessment is to gauge your mastery of the topics in this chapter. If you do not know the answer to a question or are only partially sure of the answer, you should mark that question as wrong for purposes of the self-assessment. Giving yourself credit for an answer you correctly guess skews your self-assessment results and might provide you with a false sense of security.

- 1.** For what type of production environments does Amazon recommend db.t2.small or db.t2.medium instance classes?

 - a.** Online stores
 - b.** Distributed transaction processing systems
 - c.** Bursty workloads
 - d.** Logging servers
- 2.** Which is not a TCP keepalive parameter that you can manipulate for Aurora performance?

 - a.** TCP Keepalive Time
 - b.** TCP Keepalive Priority
 - c.** TCP Keepalive Interval
 - d.** TCP Keepalive Probes
- 3.** Your RDS clients are caching their DNS data for your DB instances. What is the recommended TTL value?

 - a.** 1 day
 - b.** 1 hour
 - c.** Less than 30 seconds
 - d.** Less than 3 minutes
- 4.** Where is enhanced monitoring not available for RDS?

 - a.** US West
 - b.** US East
 - c.** East Asia
 - d.** GovCloud (US)

5. Which is not a fundamental property of your application's access patterns that you should understand when designing DynamoDB solutions?
- a. Data shape
 - b. Data currency
 - c. Data size
 - d. Data velocity
6. Which statement is false regarding adaptive capacity in DynamoDB?
- a. You must enable it if you want to use it.
 - b. Applications can continue reading and writing to hot partitions without being throttled.
 - c. Throughput capacity is automatically increased.
 - d. There can be a delay of 5 to 30 minutes prior to the activation of adaptive capacity in the system.
7. What is the term used in ElastiCache for data in the cluster that is never read?
- a. Cache slack
 - b. Cache fodder
 - c. Cache waste
 - d. Cache churn
8. Which is not a recommended best practice prior to resharding in ElastiCache?
- a. Test your application.
 - b. Get early notification for scaling issues.
 - c. Ensure sufficient free memory.

- d. Schedule resharding for baseline production windows.
- 9.** Which of the following is not a recommended best practice for Redshift performance?
 - a. Use predicates as much as possible.
 - b. Avoid cross-joins.
 - c. Use select * queries whenever possible.
 - d. Use CASE expressions whenever possible.
- 10.** What tool exists to assist you in obtaining high levels of Redshift performance?
 - a. Amazon Redshift Analyzer
 - b. Amazon Redshift Inspector
 - c. Amazon Redshift Advisor
 - d. Amazon Redshift Enhanced Monitor

FOUNDATION TOPICS

AURORA

Remember, Amazon Aurora is a MySQL-and PostgreSQL-compatible relational database built by Amazon specifically for AWS. Figure 7-1 shows Aurora in AWS.

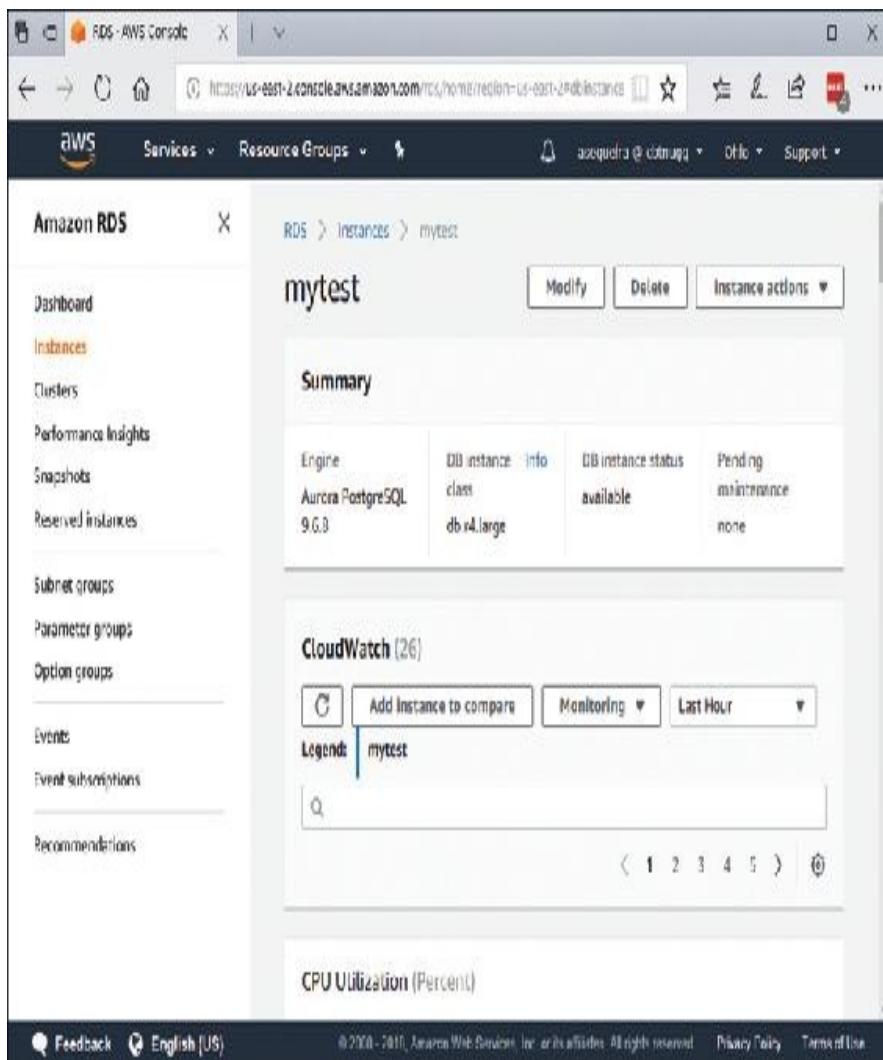


Figure 7-1 Aurora in AWS

The goal of Amazon was to combine the performance and availability of high-end commercial databases with the simplicity and cost-effectiveness of open-source databases.

Note

A more recent option with Aurora is Aurora Serverless. In this on-demand, auto-scaling configuration for Aurora (MySQL-compatible edition), the database will automatically start up, shut down, and scale capacity up or down based on your application's needs. Aurora Serverless enables you to run your database in the cloud without managing any database instances. It is a simple, cost-effective option for infrequent, intermittent, or unpredictable workloads because it automatically starts up, scales capacity to match your application's

usage, and shuts down when not in use.

The following sections describe best practices you should keep handy when working with Aurora MySQL databases.

Which DB Instance Are You Connected To?



Use the `innodb_read_only` global variable to determine which DB instance in an Aurora DB cluster you are connected to.

Here is an example:

```
show global variables like 'innodb_read_only';
```

This variable is set to ON if you are connected to an Aurora replica or OFF if you are connected to the primary instance. This value is critical to ensure that any of your write operations are using the correct connection.

When to Use T2 Instances

Aurora MySQL instances that use the `db.t2.small` or `db.t2.medium` DB instance classes are best suited for applications that do not support a high workload for an extended amount of time. Amazon recommends using the `db.t2.small` and `db.t2.medium` DB instance classes only for development and test servers or other nonproduction servers.

Note

T2 instances might be perfectly appropriate for EC2 instances in production environments.

The MySQL Performance Schema should not be enabled on Aurora MySQL T2 instances. If the Performance Schema is enabled, the T2 instance might run out of memory.

Amazon recommends the following when you use a T2 instance for the primary instance or Aurora Replicas in an Aurora MySQL DB cluster:



- If you use a T2 instance as a DB instance class in your DB cluster, use the same DB instance class for all instances in the DB cluster.
- Monitor your CPU Credit Balance (CPUCreditBalance) to ensure that it is at a sustainable level.
- When you have exhausted the CPU credits for an instance, you see an immediate drop in the available CPU and an increase in the read and write latency for the instance; this results in a severe decrease in the overall performance of the instance. If your CPU credit balance is not at a sustainable level, modify your DB instance to use one of the supported R3 DB instance classes (scale compute).
- Monitor the replica lag (AuroraReplicaLag) between the primary instance and the Aurora Replicas in the Aurora MySQL DB cluster.
- If an Aurora Replica runs out of CPU credits before the primary instance, the lag behind the primary instance results in the Aurora Replica frequently restarting. If you see a sustained increase in replica lag, make sure that your CPU credit balance for the Aurora Replicas in your DB cluster is not being exhausted.
- Keep the number of inserts per transaction below 1 million for DB clusters that have binary logging enabled.
- If the DB cluster parameter group for your DB cluster has the binlog_format parameter set to a value other than OFF, your DB cluster might experience out-of-memory conditions if the DB cluster receives transactions that contain over 1 million rows to insert. Monitor the freeable memory (FreeableMemory) metric to determine whether your DB cluster is running out of available memory.

- Check the write operations (VolumeWriteIOPS) metric to see if your primary instance is receiving a heavy load of writer operations. If this is the case, update your application to limit the number of inserts in a transaction to fewer than 1 million; alternatively, you can modify your instance to use one of the supported R3 DB instance classes (scale compute)

Work with Asynchronous Key Prefetch

Aurora can use Asynchronous Key Prefetch (AKP) to improve the performance of queries that join tables across indexes. This feature improves performance by anticipating the rows needed to run queries in which a JOIN query requires use of the Batched Key Access (BKA) Join algorithm and Multi-Range Read (MRR) optimization features.

Avoid Multithreaded Replication

By default, Aurora uses single-threaded replication when an Aurora MySQL DB cluster is used as a replication slave. While Aurora does not prohibit multithreaded replication, Aurora MySQL has inherited several issues regarding multithreaded replication from MySQL. Amazon recommends against the use of multithreaded replication in production.

Use Scale Reads

You can use Aurora with your MySQL DB instance to take advantage of the read scaling capabilities of Aurora and expand the read workload for your MySQL DB instance. To use Aurora to read scale your MySQL DB instance, create an Amazon Aurora MySQL DB cluster and make it a replication slave of your MySQL DB instance. This applies to an Amazon RDS MySQL DB instance or a MySQL database running external to Amazon RDS.

Consider Hash Joins

When you need to join a large amount of data by using an equijoin, a hash join can improve query performance. Fortunately, you can enable hash joins for Aurora MySQL. A hash join column can be any complex expression.

To find out whether a query can take advantage of a hash join, use the EXPLAIN statement to profile the query first. The EXPLAIN statement provides information about the execution plan to use for a specified query.

Use TCP Keepalive Parameters

By enabling TCP keepalive parameters and setting them aggressively, you can ensure that if your client is no longer able to connect to the database, any active connections are quickly closed. This action allows the application to react appropriately, such as by picking a new host to connect to.

The following TCP keepalive parameters need to be set:

- `tcp_keepalive_time` controls the time, in seconds, after which a keepalive packet is sent when no data has been sent by the socket (ACKs are not considered data). Amazon recommends

```
tcp_keepalive_time = 1
```

- `tcp_keepalive_intvl` controls the time, in seconds, between sending subsequent keepalive packets after the initial packet is sent (set using the `tcp_keepalive_time` parameter). Amazon recommends

```
tcp_keepalive_intvl = 1
```

- `tcp_keepalive_probes` is the number of unacknowledged keepalive probes that occur before the application is notified. Amazon recommends

```
tcp_keepalive_probes = 5
```

These settings should notify the application within 5 seconds when the database stops responding. You can set a higher `tcp_keepalive_probes` value if keepalive packets are often dropped within the application's network. This subsequently increases the time it takes to detect an actual failure but allows for more buffer in less reliable networks.

RDS

You can also experience high performance with RDS, especially if you follow some important best practices in this area. They include:



- Carefully monitor memory, CPU, and storage use with CloudWatch.
- Monitor performance metrics on a regular basis to see the average, maximum, and minimum values for a variety of time ranges.
- To be able to troubleshoot performance issues, understand the baseline performance of the system.
- Use CloudWatch events and alarms.
- Scale your DB instance as you approach capacity limits.
- Set a backup window to occur during times of traditionally low write IOPS.
- To increase the I/O capacity for a DB instance, migrate to a DB instance class with a higher I/O capacity. You can also upgrade from standard storage options, or you can provision additional throughput capacity.
- If your client application is caching the Domain Name Service (DNS) data of your DB instances, set a time-to-live (TTL) value of less than 30 seconds.
- Allocate enough RAM so that your working set resides almost completely in memory. Check the ReadIOPS metric while the DB instance is under load. The value of ReadIOPS should be small and stable.
- Because RDS provides metrics in real time for the operating system (OS)

that your DB instance runs on, view the metrics for your DB instance using the console or consume the enhanced monitoring JSON output from CloudWatch logs in a monitoring system of your choice. Enhanced monitoring is available for all DB instance classes except for db.m1.small; enhanced monitoring is available in all regions except for AWS GovCloud (US).

- ■ Tune your most commonly used and most resource-intensive queries to make them less expensive to run; this is one of the best ways to improve DB instance performance.
- ■ Try out DB parameter group changes on a test DB instance as Amazon recommends before applying parameter group changes to your production DB instances.

DYNAMODB

NoSQL database systems like DynamoDB use alternative models for data management, such as key-value pairs or document storage. It is important that you understand the key differences and the specific design approaches when you're switching from a Relational Database Management System (RDBMS) to a NoSQL database system like DynamoDB.

Figure 7-2 shows DynamoDB in AWS.

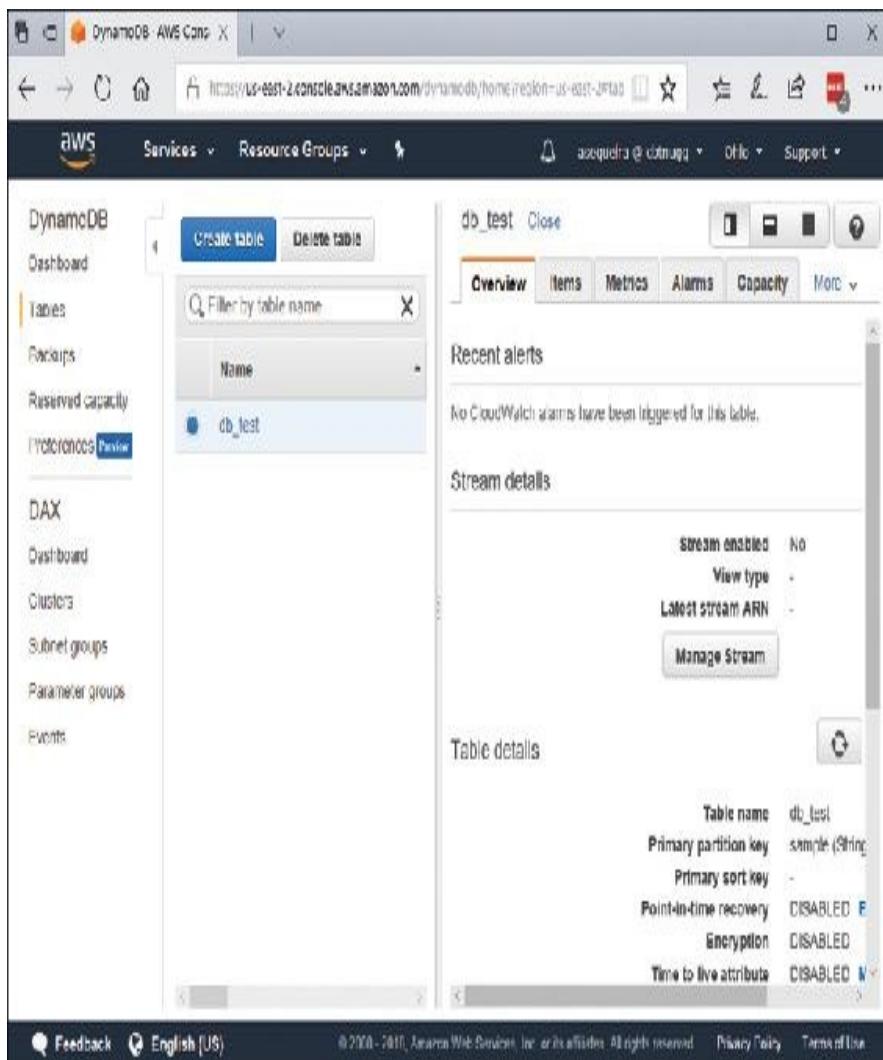


Figure 7-2 DynamoDB in AWS

In RDBMS, you design for flexibility without worrying about implementation details or performance. Query optimization generally does not affect schema design, but normalization is important. In DynamoDB, you design your schema specifically to make the most common and important queries as fast and as inexpensive as possible. Your data structures are tailored to the specific requirements of your business use cases.

For DynamoDB, you should not start designing your schema

until you know the questions it will need to answer.

Understanding the business problems and the application use cases up front is essential. You should maintain as few tables as possible in a DynamoDB application. Most well-designed applications require only one table.

The first step in designing your DynamoDB application is to identify the specific query patterns that the system must satisfy.

In particular, you need to understand three fundamental properties of your application's access patterns before you begin:

Key Topic

- **Data size:** Knowing how much data will be stored and requested at one time will help determine the most effective way to partition the data.
- **Data shape:** Instead of reshaping data when a query is processed (as an RDBMS system does), a NoSQL database organizes data so that its shape in the database corresponds with what will be queried. This is a key factor in increasing speed and scalability.
- **Data velocity:** DynamoDB scales by increasing the number of physical partitions that are available to process queries and by efficiently distributing data across those partitions. Knowing in advance what the peak query loads might be helps determine how to partition data to best use I/O capacity.

After you identify specific query requirements, you can organize data according to general principles that govern performance:

- **Keep related data together:** Research on routing-table optimization in the late 1990s found that “locality of reference” was the single most important factor in speeding up response time: keeping related data together in one place. This is equally true in NoSQL systems today, where keeping related data in close proximity has a major impact on cost and performance. Instead of distributing related data items across multiple tables, you should

keep related items in your NoSQL system as close together as possible.

- ■ **As a general rule, maintain as few tables as possible in a DynamoDB application:** Most well-designed applications require only one table, unless you have a specific reason for using multiple tables. Exceptions are cases where high-volume time series data are involved, or data sets that have very different access patterns, but these are exceptions. A single table with inverted indexes can usually enable simple queries to create and retrieve the complex hierarchical data structures that your application requires.
- ■ **Use sort order:** Related items can be grouped together and queried efficiently if their key design causes them to sort together. This is an important NoSQL design strategy.
- ■ **Distribute queries:** It is also important that a high volume of queries not be focused on one part of the database, where they can exceed I/O capacity. Instead, you should design data keys to distribute traffic evenly across partitions as much as possible, avoiding hotspots.
- ■ **Use global secondary indexes:** By creating specific global secondary indexes, you can enable different queries from those your main table can support, and that are still fast and relatively inexpensive.

These general principles translate into some common design patterns that you can use to model data efficiently in DynamoDB:

- ■ The primary key that uniquely identifies each item in a DynamoDB table can be simple (a partition key only) or composite (a partition key combined with a sort key).
- ■ Generally speaking, you should design your application for uniform activity across all logical partition keys in the table and its secondary indexes. You can determine the access patterns that your application requires and estimate the total Read Capacity Units (RCUs) and Write Capacity Units (WCUs) that each table and secondary index requires.
- ■ As traffic starts to flow, DynamoDB automatically supports your access patterns using the throughput you have provisioned, as long as the traffic against a given partition key does not exceed 3000 RCUs or 1000 WCUs.

Burst Capacity

DynamoDB provides some flexibility in your per-partition

throughput provisioning by providing burst capacity.

Whenever you are not fully using a partition’s throughput, DynamoDB reserves a portion of that unused capacity for later bursts of throughput to handle usage spikes.

DynamoDB currently retains up to 5 minutes (300 seconds) of unused read and write capacity. During an occasional burst of read or write activity, these extra capacity units can be consumed quickly—even faster than the per-second provisioned throughput capacity that you’ve defined for your table.

DynamoDB can also consume burst capacity for background maintenance and other tasks without prior notice.

Adaptive Capacity

It is not always possible to distribute read and write activity evenly all the time. When data access is imbalanced, a “hot” partition can receive a higher volume of read and write traffic compared to other partitions. In extreme cases, throttling can occur if a single partition receives more than 3000 RCU s or 1000 WCUs.

To better accommodate uneven access patterns, DynamoDB adaptive capacity enables your application to continue reading and writing to hot partitions without being throttled, provided that traffic does not exceed your table’s total provisioned capacity or the partition maximum capacity. Adaptive capacity works by automatically increasing throughput capacity for partitions that receive more traffic.

Adaptive capacity is enabled automatically for every DynamoDB table, so you do not need to explicitly enable or

disable it.

Note

Typically, a 5-to 30-minute interval occurs between the time that throttling of a hot partition begins and the time that adaptive capacity activates.

In a DynamoDB table, the primary key that uniquely identifies each item in the table can be composed not only of a partition key but also of a sort key.

Well-designed sort keys have two key benefits:

- They gather related information together in one place where it can be queried efficiently. Careful design of the sort key lets you retrieve commonly needed groups of related items using range queries with operators such as starts-with, between, >, and <.
- Composite sort keys let you define hierarchical (one-to-many) relationships in your data that you can query at any level of the hierarchy.

Secondary Indexes

Amazon DynamoDB supports two types of secondary indexes:

- **Global secondary index:** An index with a partition key and a sort key that can be different from those on the base table. A global secondary index is considered “global” because queries on the index can span all of the data in the base table, across all partitions. A global secondary index has no size limitations and has its own provisioned throughput settings for read and write activity that are separate from those of the table.
- **Local secondary index:** An index that has the same partition key as the base table but a different sort key. A local secondary index is “local” in the sense that every partition of a local secondary index is scoped to a base table partition that has the same partition key value. As a result, the total size of indexed items for any one partition key value can’t exceed 10 GB. Also, a local secondary index shares provisioned throughput settings for read and write activity with the table it is indexing.

Each table in DynamoDB is limited to a maximum of five

global secondary indexes and five local secondary indexes. For global secondary indexes, this is less restrictive than it might appear because you can satisfy multiple application access patterns with one global secondary index by overloading it.

In general, you should use global secondary indexes rather than local secondary indexes. The exception is when you need strong consistency in your query results, which a local secondary index can provide but a global secondary index cannot (global secondary index queries only support eventual consistency).

Keep the number of indexes to a minimum. Do not create secondary indexes on attributes that you do not query often. Indexes that are seldom used contribute to increased storage and I/O costs without improving application performance.

Avoid indexing tables that experience heavy write activity. In a data capture application, for example, the cost of I/O operations required to maintain an index on a table with a very high write load can be significant. If you need to index data in such a table, a more effective approach may be to copy the data to another table that has the necessary indexes and query it there.

Because secondary indexes consume storage and provisioned throughput, you should keep the size of the index as small as possible. Also, the smaller the index, the greater the performance advantage compared to querying the full table. If your queries usually return only a small subset of attributes, and the total size of those attributes is much smaller than the whole item, project only the attributes that you regularly

request.

If you expect a lot of write activity on a table compared to reads, follow these best practices:

- Consider projecting fewer attributes to minimize the size of items written to the index. However, this advice applies only if the size of projected attributes would otherwise be larger than a single write capacity unit (1 KB). For example, if the size of an index entry is only 200 bytes, DynamoDB rounds this up to 1 KB. In other words, as long as the index items are small, you can project more attributes at no extra cost.
- Avoid projecting attributes that you know will rarely be needed in queries. Every time you update an attribute that is projected in an index, you incur the extra cost of updating the index as well. You can still retrieve nonprojected attributes in a query at a higher provisioned throughput cost, but the query cost may be significantly lower than the cost of updating the index frequently.
- Specify ALL only if you want your queries to return the entire table item sorted by a different sort key. Projecting all attributes eliminates the need for table fetches, but in most cases, it doubles your costs for storage and write activity.
- To get the fastest queries with the lowest possible latency, project all the attributes that you expect those queries to return. In particular, if you query a local secondary index for attributes that are not projected, DynamoDB automatically fetches those attributes from the table, which requires reading the entire item from the table. This introduces latency and additional I/O operations that you can avoid.
- When you create a local secondary index, think about how much data will be written to it and how many of those data items will have the same partition key value. If you expect that the sum of table and index items for a particular partition key value might exceed 10 GB, consider whether you should avoid creating the index.
- If you cannot avoid creating the local secondary index, anticipate the item collection size limit and take action before you exceed it.

For any item in a table, DynamoDB writes a corresponding index entry only if the index sort key value is present in the item. If the sort key does not appear in every table item, the

index is said to be sparse. Sparse indexes are useful for queries over a small subsection of a table.

Global secondary indexes are sparse by default. When you create a global secondary index, you specify a partition key and optionally a sort key. Only items in the parent table that contain those attributes appear in the index. By designing a global secondary index to be sparse, you can provision it with lower write throughput than that of the parent table, while still achieving excellent performance.

Querying and Scanning Data

In general, scan operations are less efficient than other operations in DynamoDB. A scan operation always scans the entire table or secondary index. It then filters out values to provide the result you want, adding the extra step of removing data from the result set. If possible, you should avoid using a scan operation on a large table or index with a filter that removes many results. Also, as a table or index grows, the scan operation slows. The scan operation examines every item for the requested values and can use up the provisioned throughput for a large table or index in a single operation. For faster response times, design your tables and indexes so that your applications can use query instead of scan. (For tables, you can also consider using the GetItem and BatchGetItem APIs.)

Alternatively, design your application to use scan operations in a way that minimizes the impact on your request rate.

When you create a table, you set its read and write capacity unit requirements. For reads, the capacity units are expressed as the number of strongly consistent 4 KB data read requests

per second. For eventually consistent reads, a read capacity unit is two 4 KB read requests per second. A scan operation performs eventually consistent reads by default, and it can return up to 1 MB (one page) of data. Therefore, a single scan request can consume (1 MB page size 4 KB item size) 2^{10} (eventually consistent reads) = 128 read operations. If you request strongly consistent reads instead, the scan operation would consume twice as much provisioned throughput—256 read operations. This represents a sudden spike in usage, compared to the configured read capacity for the table. This usage of capacity units by a scan prevents other potentially more important requests for the same table from using the available capacity units.

As a result, you likely get a ProvisionedThroughputExceeded exception for those requests. The problem is not just the sudden increase in capacity units that the scan uses. The scan is also likely to consume all of its capacity units from the same partition because the scan requests read items that are next to each other on the partition. This means that the request is hitting the same partition, causing all of its capacity units to be consumed, and throttling other requests to that partition. If the request to read data is spread across multiple partitions, the operation would not throttle a specific partition.

Because a scan operation reads an entire page (by default, 1 MB), you can reduce the impact of the scan operation by setting a smaller page size. The scan operation provides a Limit parameter that you can use to set the page size for your request. Each query or scan request that has a smaller page size uses fewer read operations and creates a “pause” between each request. For example, suppose that each item is 4 KB and

you set the page size to 40 items. A query request would then consume only 20 eventually consistent read operations or 40 strongly consistent read operations. A larger number of smaller query or scan operations would allow your other critical requests to succeed without throttling.

Isolate scan operations; DynamoDB is designed for easy scalability. As a result, an application can create tables for distinct purposes, possibly even duplicating content across several tables. You want to perform scans on a table that is not taking mission-critical traffic. Some applications handle this load by rotating traffic hourly between two tables: one for critical traffic and one for bookkeeping. Other applications can do this by performing every write on two tables: a mission-critical table and a shadow table.

Configure your application to retry any request that receives a response code that indicates you have exceeded your provisioned throughput. Or, increase the provisioned throughput for your table using the `UpdateTable` operation. If you have temporary spikes in your workload that cause your throughput to exceed, occasionally, beyond the provisioned level, retry the request with exponential backoff.

Many applications can benefit from using parallel scan operations rather than sequential scans. For example, an application that processes a large table of historical data can perform a parallel scan much faster than a sequential one. Multiple worker threads in a background “sweeper” process could scan a table at a low priority without affecting production traffic. In each of these examples, a parallel scan is used in such a way that it does not starve other applications of

provisioned throughput resources.

Although parallel scans can be beneficial, they can place a heavy demand on provisioned throughput. With a parallel scan, your application has multiple workers that are all running scan operations concurrently. This can quickly consume all of your table's provisioned read capacity. In that case, other applications that need to access the table might be throttled.

A parallel scan can be the right choice if the following conditions are met:

- ■ The table size is 20 GB or larger.
- ■ The table's provisioned read throughput is not being fully used.
- ■ Sequential scan operations are too slow.

The best setting for TotalSegments depends on your specific data, the table's provisioned throughput settings, and your performance requirements. You might need to experiment to get it right.

ELASTICACHE

Although you might assume that simply using ElastiCache provides performance improvements just through its usage, there are some key best practices to making the most of its performance.



Let's begin with caching strategies. The strategy or strategies you want to implement for populating and maintaining your cache depend on what data you're caching and the access patterns to that data. For example, you likely would not want

to use the same strategy for a Top-10 leaderboard on a gaming site, Facebook posts, and trending news stories. In the following sections, we discuss common cache maintenance strategies, their advantages, and their disadvantages.

Lazy Loading Versus Write Through

ElastiCache is an in-memory key-value store that sits between your application and the data store (database) that it accesses. Whenever your application requests data, it first makes the request to the ElastiCache cache. If the data exists in the cache and is current, ElastiCache returns the data to your application. If the data does not exist in the cache, or the data in the cache has expired, your application requests the data from your data store, which returns the data to your application. Your application then writes the data received from the store to the cache so it can be more quickly retrieved the next time it is requested.

Scenario 1: Cache Hit

When data is in the cache and isn't expired:

- Application requests data from the cache.
- Cache returns the data to the application.

Scenario 2: Cache Miss

When data isn't in the cache or is expired:

- Application requests data from the cache.
- Cache doesn't have the requested data, so it returns a null.
- Application requests and receives the data from the database.
- Application updates the cache with the new data.

Advantages and Disadvantages of Lazy Loading

Table 7-2 presents the advantages and disadvantages of lazy

loading.

Table 7-2 Advantages and Disadvantages of Lazy Loading

Advantages	Disadvantages
Only requested data is cached.	There is a cache miss penalty. Each cache miss results in three trips: the initial request for data from the cache; the query of the database for the data; and the writing of data to the cache, which can cause a noticeable delay in data getting to the application.
Because most data is never requested, lazy loading avoids filling up the cache with data that is not requested.	If data is written to the cache only when there is a cache miss, data in the cache can become <i>stale</i> because there are no updates to the cache when data is changed in the database. This issue is addressed by the write through and adding TTL strategies.
Node failures are not fatal.	

Advantages and Disadvantages of Write Through

An alternative approach to lazy loading is write through. Table 7-3 presents the advantages and disadvantages of write through.

Table 7-3 Advantages and Disadvantages of Write Through

Advantages	Disadvantages
Data in the cache is never stale: Because the data in the cache is updated every time it is written to the database, the data in the cache is always current.	Missing data: In the case of spinning up a new node, whether due to a node failure or scaling out, there is missing data that continues to be missing until it is added or updated on the database. This situation can be minimized by implementing lazy loading in conjunction with write through.
Write penalty vs. read penalty: Every write involves two trips: a write to the cache and a write to the database. This adds latency to the process. That said, end users are generally more tolerant of latency when updating data than when retrieving data. There is an inherent sense that updates are more work and thus take longer.	Cache churn: Because most data is never read, a lot of data in the cluster is never read. This is a waste of resources. By adding TTL, you can minimize wasted space.

Lazy loading allows for stale data but does not fail with empty nodes. Write through ensures that data is always fresh but may fail with empty nodes and may populate the cache with superfluous data. By adding a time-to-live (TTL) value to each write, you are able to enjoy the advantages of each strategy and largely avoid cluttering up the cache with superfluous data.

What is TTL?

WHAT IS TTL?

Time to live (TTL) is an integer value that specifies the number of seconds (or milliseconds) until the key expires.

When an application attempts to read an expired key, it is treated as though the key is not found, meaning that the database is queried for the key and the cache is updated. This does not guarantee that a value is not stale, but it keeps data from getting too stale and requires that values in the cache are occasionally refreshed from the database.

Background Write Process and Memory Usage

Whenever a background write process is called, Redis forks its process (remember, Redis is single threaded). One fork persists your data to disk in a Redis .rdb snapshot file. The other fork services all read and write operations. To ensure that your snapshot is a point-in-time snapshot, all data updates and additions are written to an area of available memory separate from the data area.

As long as you have sufficient memory available to record all write operations while the data is being persisted to disk, you should have no insufficient memory issues. You are likely to experience insufficient memory issues if any of the following are true:

- ■ Your application performs many write operations, thus requiring a large amount of available memory to accept the new or updated data.
- ■ You have very little memory available in which to write new or updated data.
- ■ You have a large data set that takes a long time to persist to disk, thus requiring a large number of write operations.

Avoid Running Out of Memory When Executing a Background Write

Whenever a background write process such as BGSAVE or BGREWRITEAOF is called, to keep the process from failing, you must have more memory available than will be consumed by write operations during the process. The worst-case scenario is that during the background write operation every Redis record is updated and some new records are added to the cache. Because of this, we recommend that you set reserved-memory-percent to 50 (50 percent) for Redis versions before 2.8.22 or 25 (25 percent) for Redis versions 2.8.22 and later.

The maxmemory value indicates the memory available to you for data and operational overhead. Because you cannot modify the reserved-memory parameter in the default parameter group, you must create a custom parameter group for the cluster. The default value for reserved-memory is 0, which allows Redis to consume all of maxmemory with data, potentially leaving too little memory for other uses, such as a background write process.

You can also use reserved-memory parameter to reduce the amount of memory Redis uses on the box.

How Much Reserved Memory Do You Need?

If you are running a version of Redis prior to 2.8.22, you need to reserve more memory for backups and failovers than if you are running Redis 2.8.22 or later. This requirement is due to the different ways that ElastiCache for Redis implements the backup process. The guiding principle is to reserve half of a node type's maxmemory value for Redis overhead for versions prior to 2.8.22 and one-fourth for Redis versions 2.8.22 and later.

Parameters to Manage Reserved Memory

As of March 16, 2017, Amazon ElastiCache for Redis provides two mutually exclusive parameters for managing your Redis memory: reserved-memory and reserved-memory-percent. Neither one of these parameters is part of the Redis distribution. Depending on when you became an ElastiCache customer, one or the other of these parameters is the default memory management parameter whenever you create a new Redis cluster or replication group and use a default parameter group. If you were already a customer as of March 16, 2017, whenever you create a Redis cluster or replication group using the default parameter group, your memory management parameter is reserved-memory with zero (0) bytes of memory reserved. If you became a customer on or after March 16, 2017, whenever you create a Redis cluster or replication group using the default parameter group, your memory management parameter is reserved-memory-percent with 25 percent of your node's maxmemory reserved for nondata purposes.

If, after reading about the two Redis memory management parameters, you prefer to use the one that isn't your default or with nondefault values, you can change to the other reserved memory management parameter. If you later need to change the value of that parameter, you can create a custom parameter group and modify it to use your preferred memory management parameter and value. You can then use the custom parameter group whenever you create a new Redis cluster or replication group. For existing clusters or replication groups, you can modify them to use your custom parameter group.

Prior to March 16, 2017, all ElastiCache for Redis reserved memory management was done using the parameter reserved-

memory. The default value of reserved-memory is 0. This default reserves no memory for Redis overhead and allows Redis to consume all of a node's memory with data. Changing reserved-memory so you have sufficient memory available for backups and failovers requires you to create a custom parameter group. In this custom parameter group, you set reserved-memory to a value appropriate for the Redis version running on your cluster and cluster's node type.

The ElastiCache for Redis parameter reserved-memory is specific to ElastiCache for Redis and isn't part of the Redis distribution.

Online Cluster Resizing

Resharding involves adding and removing shards or nodes to your cluster and redistributing key spaces. As a result, multiple things have an impact on the resharding operation, such as the load on the cluster, memory utilization, and overall size of data. For the best experience, we recommend that you follow overall cluster best practices for uniform workload pattern distribution. In addition, we recommend taking the following steps.

Before initiating resharding, we recommend the following:

- **Test your application:** Test your application behavior during resharding in a staging environment if possible.
- **Get early notification for scaling issues:** Because resharding is a compute-intensive operation, we recommend keeping CPU utilization under 80 percent on multicore instances and less than 50 percent on single core instances during resharding. Monitor ElastiCache for Redis metrics and initiate resharding before your application starts observing scaling issues. Useful metrics to track are CPUUtilization, NetworkBytesIn, NetworkBytesOut, CurrConnections, NewConnections, FreeableMemory, SwapUsage, and BytesUsedForCache.

- **■ Ensure sufficient free memory is available before scaling in:** If you're scaling in, ensure that free memory available on the shards to be retained is at least 1.5 times the memory used on the shards you plan to remove.
- **■ Initiate resharding during off-peak hours:** This practice helps to reduce the latency and throughput impact on the client during the resharding operation. It also helps to complete resharding faster as more resources can be used for slot redistribution.
- **■ Review client timeout behavior:** Some clients might observe higher latency during online cluster resizing. Configuring your client library with a higher timeout can help by giving the system time to connect even under higher load conditions on the server. If you open a large number of connections to the server, consider adding exponential backoff to reconnect logic to prevent a burst of new connections hitting the server at the same time.

During resharding, the following practices are recommended:

- **■ Avoid expensive commands:** Avoid running any computationally and I/O intensive operations, such as the KEYS and SMEMBERS commands. We suggest this approach because these operations increase the load on the cluster and have an impact on the performance of the cluster. Instead, use the SCAN and SSCAN commands.
- **■ Follow Lua best practices:** Avoid long-running Lua scripts and always declare keys used in Lua scripts up front. We recommend this approach to determine that the Lua script is not using cross slot commands. Ensure that the keys used in Lua scripts belong to the same slot.

After resharding, note the following:

- **■ Scale-in might be partially successful if insufficient memory is available on target shards.** If such a result occurs, review available memory and retry the operation, if necessary.
- **■ Slots with large items are not migrated.** In particular, slots with items larger than 256 MB postserialization are not migrated.
- **■ The BRPOPLPUSH command is not supported if it operates on the slot being migrated.** FLUSHALL and FLUSHDB commands are not supported inside Lua scripts during a resharding operation.

REDSHIFT

This section presents best practices for designing tables, loading data into tables, and writing queries for Amazon Redshift, and also a discussion of working with [Amazon Redshift Advisor](#).

Amazon Redshift is not the same as other SQL database systems. To fully realize the benefits of the Amazon Redshift architecture, you must specifically design, build, and load your tables to use massively parallel processing, columnar data storage, and columnar data compression. If your data loading and query execution times are longer than you expect, or longer than you want, you might be overlooking key information.

Note

Amazon Redshift Spectrum allows you to store data in Amazon S3, in open file formats, and have it available for analytics without the need to load it into your Amazon Redshift cluster. It enables you to easily join data sets across Redshift clusters and S3 to provide unique insights that you would not be able to obtain by querying independent data silos.

As you plan your database, certain key table design decisions heavily influence overall query performance. These design choices also have a significant effect on storage requirements, which in turn affect query performance by reducing the number of I/O operations and minimizing the memory required to process queries.

Consider the following best practices:

- ■ Use a COPY command to load data.
- ■ Use a single COPY command to load from multiple files.
- ■ Split your load data into multiple files.
- ■ Compress your data files.

- ■ Use a manifest file.
- ■ Verify data files before and after a load.
- ■ Use a multi-row insert.
- ■ Use a bulk insert.
- ■ Load data in sort key order.
- ■ Load data in sequential blocks.
- ■ Use time-series tables.
- ■ Use a staging table to perform a merge (Upsert).
- ■ Schedule around maintenance windows.

Amazon Redshift Best Practices for Designing Queries

To maximize query performance, follow these recommendations when creating queries:



- ■ Design tables according to best practices to provide a solid foundation for query performance.
- ■ Avoid using select *. Include only the columns you specifically need.
- ■ Use a CASE expression to perform complex aggregations instead of selecting from the same table multiple times.
- ■ Don't use cross-joins unless absolutely necessary. These joins without a join condition result in the Cartesian product of two tables. Cross-joins are typically executed as nested-loop joins, which are the slowest of the possible join types.
- ■ Use subqueries in cases where one table in the query is used only for predicate conditions and the subquery returns a small number of rows (less than about 200).
- ■ Use predicates to restrict the data set as much as possible. In the predicate, use the least expensive operators that you can. Comparison Condition operators are preferable to LIKE operators. LIKE operators are still preferable to SIMILAR TO or POSIX operators.

- ■ Avoid using functions in query predicates. Using them can drive up the cost of the query by requiring large numbers of rows to resolve the intermediate steps of the query.
- ■ If possible, use a WHERE clause to restrict the data set. The query planner can then use row order to help determine which records match the criteria, so it can skip scanning large numbers of disk blocks. Without this, the query execution engine must scan participating columns entirely.
- ■ Add predicates to filter tables that participate in joins, even if the predicates apply the same filters. The query returns the same result set, but Amazon Redshift is able to filter the join tables before the scan step and can then efficiently skip scanning blocks from those tables. Redundant filters aren't needed if you filter on a column that's used in the join condition.
- ■ Use sort keys in the GROUP BY clause so the query planner can use more efficient aggregation. A query might qualify for one-phase aggregation when its GROUP BY list contains only sort key columns, one of which is also the distribution key. The sort key columns in the GROUP BY list must include the first sort key, then other sort keys that you want to use in sort key order. For example, it is valid to use the first sort key; the first and second sort keys; the first, second, and third sort keys; and so on. It is not valid to use the first and third sort keys.

Work with Recommendations from Amazon Redshift Advisor

To help you improve the performance and decrease the operating costs for your Redshift cluster, Redshift Advisor offers you specific recommendations about changes to make. Advisor develops its customized recommendations by analyzing performance and usage metrics for your cluster. These tailored recommendations relate to operations and cluster settings. To help you prioritize your optimizations, Advisor ranks recommendations by order of impact.

Advisor bases its recommendations on observations regarding performance statistics or operations data. Advisor develops observations by running tests on your clusters to determine if a

test value is within a specified range. If the test result is outside of that range, Advisor generates an observation for your cluster. At the same time, Advisor creates a recommendation about how to bring the observed value back into the best-practice range. Advisor displays only recommendations that will have a significant impact on performance and operations. When Advisor determines that a recommendation has been addressed, it removes it from your recommendation list. For example, if your data warehouse contains a large number of uncompressed table columns, you can save on cluster storage costs by rebuilding tables using the ENCODE parameter to specify column compression. If the Advisor observes that your cluster contains a significant amount of data in uncompressed table data, it provides you with the SQL code block to find the table columns that are candidates for compression and resources that describe how to compress those columns.

Advisor recommendations are made up of the following sections:

- **■ Best practice recommendation:** This is a brief summary of the recommendation.
- **■ Observation:** Findings from tests run on your cluster to determine if a test value is within a specified range.
- **■ Recommendations:** These recommendations provide specific steps to take and implementation tips.
- **■ Provide feedback:** Your feedback can be detailed and goes directly to the Amazon Redshift engineering team.

EXAM PREPARATION TASKS

As mentioned in the section “How to Use This Book” in the Introduction, you have a couple of choices for exam preparation: the exercises here, Chapter 16, “Final Preparation,” and the exam simulation questions in the Pearson Test Prep practice test software.

REVIEW ALL KEY TOPICS

Review the most important topics in this chapter, noted with the Key Topics icon in the margin of the page. Table 7-4 lists a reference of these key topics and the page numbers on which each is found.

Table 7-4 Key Topics for Chapter 7

 Key Topic Element	Description	Page Number
Overview	Best practices for Aurora MySQL databases	115
List	T2 instance best practices for Aurora	115
List	RDS performance best practices	118

List	DynamoDB query pattern properties	120
Text	Caching strategies	127
List	Redshift query design best practices	133

COMPLETE TABLES AND LISTS FROM MEMORY

Print a copy of [Appendix C, “Memory Tables”](#) (found on the book website), or at least the section for this chapter, and complete the tables and lists from memory. [Appendix D, “Memory Tables Answer Key,”](#) also on the website, includes completed tables and lists to check your work.

DEFINE KEY TERMS

Define the following key terms from this chapter and check your answers in the glossary:

AKP

Global secondary index

Local secondary index

Lazy loading

Write through

Amazon Redshift Advisor

Q&A

The answers to these questions appear in [Appendix A](#). For more practice with exam format questions, use the Pearson Test Prep practice test software.

- 1.** How would you implement scale reads in Amazon Aurora?
- 2.** What are three options for increasing the I/O capacity of your DB instance in RDS?
- 3.** What is the general recommendation on the number of tables you should use in a DynamoDB design?
- 4.** Cache churn is the result of what approach to caching with a system like ElastiCache?
- 5.** How does the Amazon Redshift Advisor know to make recommendations to you?

Chapter 8. Improving Performance with Caching

This chapter covers the following subjects:

- **ElastiCache:** This powerful service allows you to cache data in memory so that it can be retrieved quickly and efficiently for your customers.
- **DynamoDB Accelerator:** This section introduces the DynamoDB Accelerator (DAX) technology that provides specialized caching tuned for DynamoDB.
- **CloudFront:** Being able to deliver cached copies of your content that is geographically close to your users would be excellent. You can do this thanks to CloudFront. In this section, you learn about CloudFront and even set up your own distribution.
- **Greengrass:** While not strictly a caching service, Greengrass is close enough to get a mention in this chapter. AWS Greengrass is for IoT implementations when you want to run Lambda functions locally on remote IoT devices.
- **Route 53:** Caching plays a massive role in improving DNS through Route 53 performance. This section breaks all this down for you.

Caching has always been an excellent way to improve computing and networking performance. When data that you need already exists in a system's memory resources, you can design for blazing application performance. This chapter focuses on various caching services and approaches in AWS.

“DO I KNOW THIS ALREADY?” QUIZ

The “Do I Know This Already?” quiz allows you to assess whether you should read the entire chapter. Table 8-1 lists the major headings in this chapter and the “Do I Know This Already?” quiz questions covering the material in those

headings so you can assess your knowledge of these specific areas. The answers to the “Do I Know This Already?” quiz appear in [Appendix A](#).

Table 8-1 “Do I Know This Already?” Foundation Topics Section-to-Question Mapping

Foundation Topics Section	Questions
ElastiCache	1–2
DynamoDB Accelerator	3–4
CloudFront	5–6
Greengrass	7
Route 53	8

Caution

The goal of self-assessment is to gauge your mastery of the topics in this chapter. If you do not know the answer to a question or are only partially sure of the answer, you should mark that question as wrong for purposes of the self-assessment. Giving yourself credit for an answer

you correctly guess skews your self-assessment results and might provide you with a false sense of security.

- 1.** What services is ElastiCache often used with? (Choose two.)
 - a.** DynamoDB
 - b.** Route 53
 - c.** CloudFront
 - d.** RDS

- 2.** What engine is supported by ElastiCache and offers the option of a multi-AZ deployment?
 - a.** memcached
 - b.** Greengrass
 - c.** Redis
 - d.** Hadoop

- 3.** DAX is an accelerator for which product?
 - a.** Glacier
 - b.** Redshift
 - c.** RDS
 - d.** DynamoDB

- 4.** Which database would DAX not be appropriate for?
 - a.** One that features repeated reads for individual keys
 - b.** One that is bursty in nature
 - c.** One that is write intensive
 - d.** One that is read intensive

- 5.** What are the sources called in CloudFront?

- a. Edge devices
 - b. Content servers
 - c. Distribution servers
 - d. Origins
- 6.** What is the default amount of time a resource will remain cached in an edge location with CloudFront?

 - a. 3 minutes
 - b. 24 hours
 - c. 1 hour
 - d. 30 minutes
- 7.** Greengrass is used to run code locally on what types of systems?

 - a. On-prem databases
 - b. On-prem IoT devices
 - c. On-prem LUNs
 - d. On-prem web servers
- 8.** What value in Route 53 do you use to control the caching behavior on DNS resolvers?

 - a. Minimum visibility timeout
 - b. Resolver cache duration
 - c. Keepalive
 - d. TTL

FOUNDATION TOPICS

FI ΔΣΤΙΚΑΣ ΤΗΣ

ELASTICACHE

It's hard not to think of the ElastiCache service when you hear caching mentioned. This is indeed one of the primary in-memory caching services for your AWS cloud. Thanks to ElastiCache, various forms of data that your clients might need through their applications can be served up quickly from memory on an AWS system instead of being called from a disk-based database system. Although this book teaches you how to build performant database implementations, there is nothing faster than retrieving data from memory as opposed to a retrieving it from a traditional database system.

Although this service is not restricted for use with databases, it is often used in conjunction with either DynamoDB or RDS database services of AWS. Imagine a scenario where you have a complex query containing a bunch of data that is popular for your clients to receive. It would make good sense to cache this query result so that you can provide faster access for your clients.



ElastiCache supports two of the most popular caching engine technologies. There is support for Redis and memcached. memcached is a simple service, whereas Redis provides support for more complex data types and storage approaches. Redis also supports multi-AZ deployments inside AWS.

Lab: Configuring a Redis ElastiCache Cluster

In the following steps, you configure an ElastiCache implementation. You set up the cache and engine and then test this cache by connecting to it and running some sample

commands.

Step 1. From the AWS Management Console, search for **ElastiCache** and select the link.

Step 2. Click **Get Started Now** to configure ElastiCache.

Step 3. Ensure that Redis is selected for the cluster engine.

Name your cluster engine **myrediscache** and choose the node type **cache.t2.micro**. Set the number of replicas to **None**. Hide the Advanced Redis Settings area because you don't need these settings in this lab. Figure 8-1 shows this configuration.

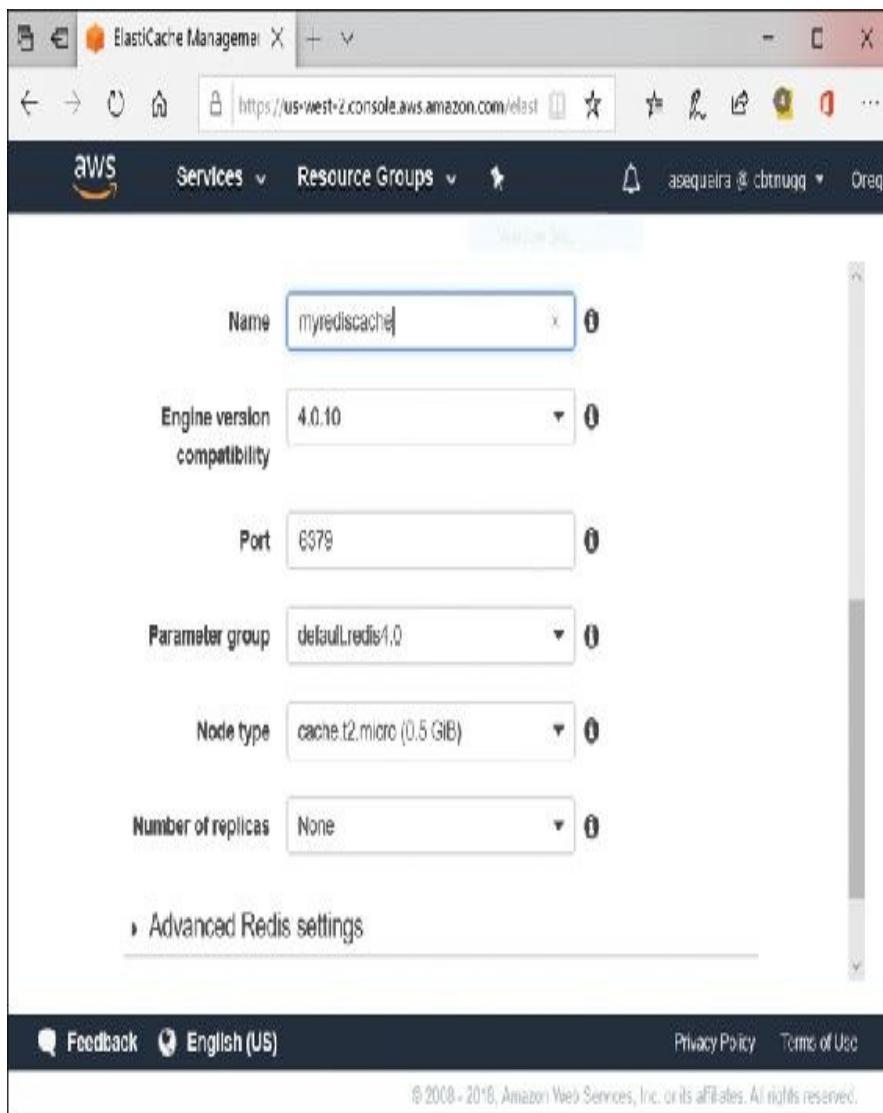


Figure 8-1 Configuring a Redis ElastiCache Cluster

Step 4. Click the **Create** button near the bottom of the page.

Step 5. Wait for the status of your cluster to change from Creating... to Available. This should take about 10 minutes.

Step 6. Create a Windows Server 2016 EC2 instance for testing your Redis ElastiCache Cluster. To do so, use the following parameters:

- ■ AMI: **Microsoft Windows Server 2016 Base-ami**
- ■ Instance: **t2.micro**
- ■ Instance Details: **Defaults**
- ■ Storage: **Defaults**
- ■ Add Tags: **Name – MyTestSystem**
- ■ Configure Security Group: **New – MyTestGroup**
- ■ Security Group Rules: Add the TCP Protocol port **6379** for Redis

Step 7. Connect to your test system using RDP.

Step 8. Set up the browser in Windows Server to easily navigate to websites. To do so, click the **Start** button and then click **Server Manager**. Select **Local Server** in the left navigation pane and then use the links on the right to turn off the IE Enhanced Security Configuration for all users.

Step 9. Open the web browser and navigate to
<https://redisdesktop.com/download>.

Step 10. Download and install the Redis Desktop Manager client using the defaults for installation.

Step 11. To obtain the endpoint name for the Redis ElastiCache cluster, navigate to the ElastiCache service.

Step 12. Select the **Redis** link to view your cluster.

Step 13. Select the drop-down option to see the details about your cluster, as shown in Figure 8-2.

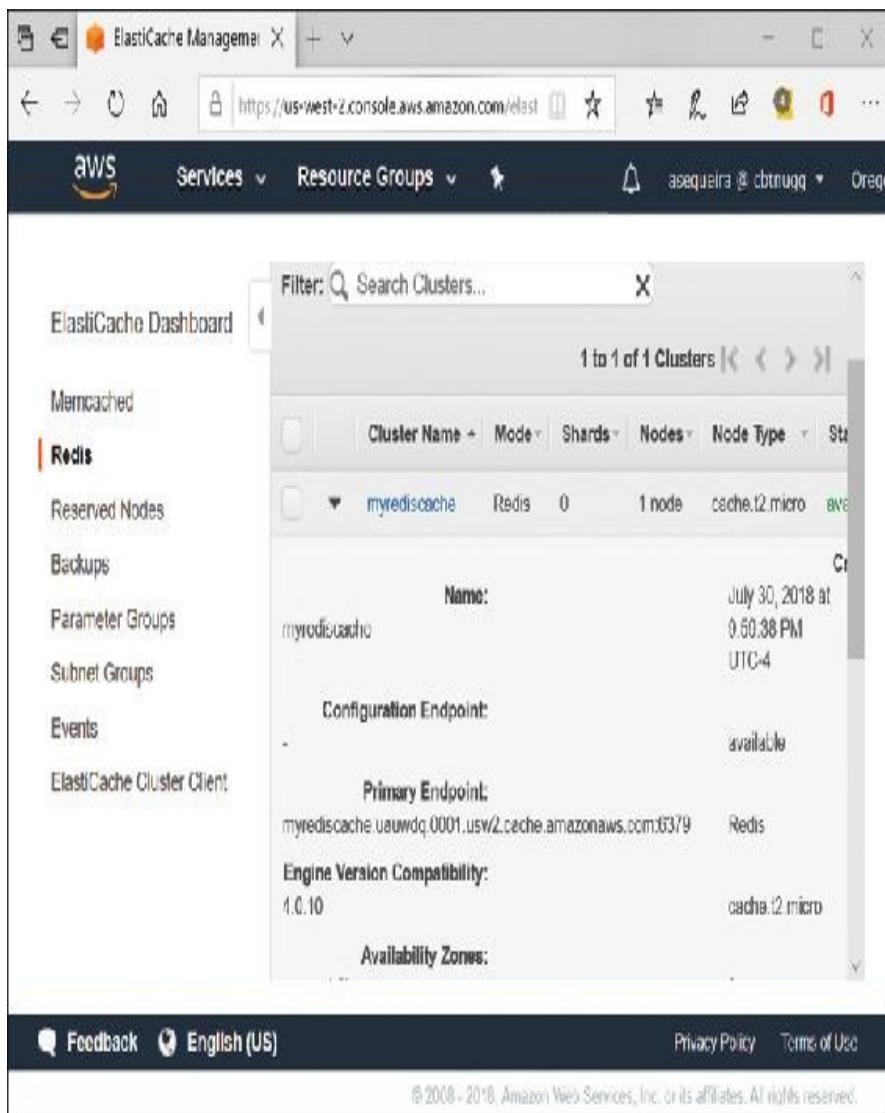


Figure 8-2 Examining the Redis Cluster Details

Step 14. Copy the primary endpoint text (without the port number portion :6379).

Step 15. Modify the security group for the cluster to match the one created for the test server. To do so, click the check box next to your cluster name and then click the **Modify** button. Edit the VPC Security Group and choose **MyTestGroup**. Click the **Modify** button. Now wait until your cluster status is

Available before moving on to the next step.

Step 16. In your test Windows Server system, launch the Redis Desktop Manager and choose **Connect to Redis Server** from the top menu.

Step 17. Paste your primary endpoint address in the Address field and name the connection **Test**. Click **OK** to make your connection to ElastiCache.

Step 18. Click your **Test** connection and click the **Open Console** button to make a console connection to the Redis cluster.

Step 19. Click in the console and try some Redis commands:

```
set a hello
get a
memory malloc-stats
```

Lab Cleanup

Step 1. In the ElastiCache service area, select **myrediscache** and click the **Delete** button.

Step 2. Navigate to EC2. Click **Running Instances** and choose the **MyTestSystem** instance. Click **Actions**, then **Instance State**, and then **Terminate**. Click the **Yes, Terminate** button.

Step 3. Wait some time for your system to terminate. Click **Security Groups** in the left navigation links.

Step 4. Select the **MyTestGroup** security group and click **Actions** and then **Delete**. If you receive an error

message that the security group cannot be deleted, you have not waited long enough for the test machine termination.

DYNAMODB ACCELERATOR

Although ElastiCache usage with DynamoDB was a frequent solution for many AWS architects, another option is designed specifically for DynamoDB caching. The name of this service is DynamoDB Accelerator (DAX).

The AWS DynamoDB service is designed for scale and performance. In most cases, the DynamoDB response times can be measured in single-digit milliseconds. However, certain use cases require response times in microseconds. For these use cases, DynamoDB Accelerator delivers fast response times for accessing eventually consistent data.

DAX is a DynamoDB-compatible caching service that enables you to benefit from fast in-memory performance for demanding applications. DAX addresses three core scenarios:



- As an in-memory cache, DAX reduces the response times of eventually consistent read workloads by an order of magnitude from single-digit milliseconds to microseconds.
- DAX reduces operational and application complexity by providing a managed service that is API-compatible with Amazon DynamoDB and thus requires only minimal functional changes to use with an existing application.
- For read-heavy or bursty workloads, DAX provides increased throughput and potential operational cost savings by reducing the need to overprovision read capacity units. This capability is especially beneficial for applications that require repeated reads for individual keys.

DAX is not always an ideal solution. For example, DAX

would not be appropriate under these conditions:

- ■ For applications that require strongly consistent reads (or cannot tolerate eventually consistent reads)
- ■ For applications that do not require microsecond response times for reads or that do not need to offload repeated read activity from underlying tables
- ■ For applications that are write-intensive or that do not perform much read activity
- ■ For applications that are already using a different caching solution with DynamoDB and are using their own client-side logic for working with that caching solution

DAX functions thanks to the following components:

- ■ **Nodes:** The smallest building blocks of DAX. Each node contains a single replica of the cached data for DynamoDB. You can scale your DAX cluster by adding nodes to the cluster or by using larger node types or both.
- ■ **Cluster:** A logical grouping of one or more nodes. If there is more than one node, there is a primary node and *read replicas*. A cluster can support up to ten nodes, and Amazon recommends a minimum of three nodes in a cluster with each node placed in a different Availability Zone.

CLOUDFRONT

CloudFront speeds up distribution of your content to your end users. It does this by delivering content from a worldwide network of data centers called *edge locations*. As with all caching solutions, the content might not be available in the cache for an end user (cache miss). When this happens, CloudFront can retrieve the content from an origin that you have defined. This location might be an S3 bucket or a web server that you have identified as the source of your content.

As described previously, you configure CloudFront as follows:

Key Topic

Step 1. You specify an origin from which CloudFront will get your files for distribution.

Step 2. You upload the files to your origin.

Step 3. You create a CloudFront distribution. This tells CloudFront which origin to get your files from. You also set details here such as whether you want requests logged.

Step 4. CloudFront assigns a domain name to your new distribution.

Step 5. CloudFront configures its edge locations with your distribution configuration.

As you develop your websites and/or applications, you can use the domain name provided by CloudFront, or you can configure your own domain to redirect to the CloudFront domain. You can also configure your origin server to add headers to your CloudFront content. These headers can indicate how long each object is to remain cached in an edge location. By default, each object will remain in the edge location for 24 hours before expiration. There is no upper limit on this duration.

Lab: Configuring CloudFront

In the the following steps, you make an image file from an S3 bucket available via CloudFront:

Step 1. Open the AWS Management Console and navigate to the **S3** service.

Step 2. In S3, choose **Create Bucket** and give your bucket your last name with four random numbers—for

example, in my case, sequeira1024. Click **Create**.

Step 3. Click on your bucket name to enter it. Click **Upload** and click **Add Files** to upload a JPG file from your local computer. If you do not have a JPG handy, visit a website, right-click an image, and choose **File Save As**. After you select a JPG, click **Next**.

Step 4. Under Manage Public Permissions, choose **Grant Public Read Access to This Object(s)** and click **Upload**.

Step 5. Click the name of your JPG in the S3 bucket to access its properties. Copy the link to your file and paste it in a web browser to ensure you can access it from the S3 bucket via the Internet.

Step 6. Use the Management Console to search for the CloudFront service and click the link.

Step 7. Click **Create Distribution**.

Step 8. In the Select a Delivery Method for Your Content Area in the Web section, click **Get Started**.

Step 9. Click in the **Origin Domain Name** field and choose the S3 bucket you created in this lab.

Step 10. Examine the many options you can configure and then choose **Create Distribution** near the bottom of the page to accept the defaults.

Step 11. Wait 15 to 20 minutes for the status of your CloudFront distribution to change to Deployed.

Step 12. Click on the **ID** link of your CloudFront

distribution to view the domain name of your distribution, as shown in [Figure 8-3](#).

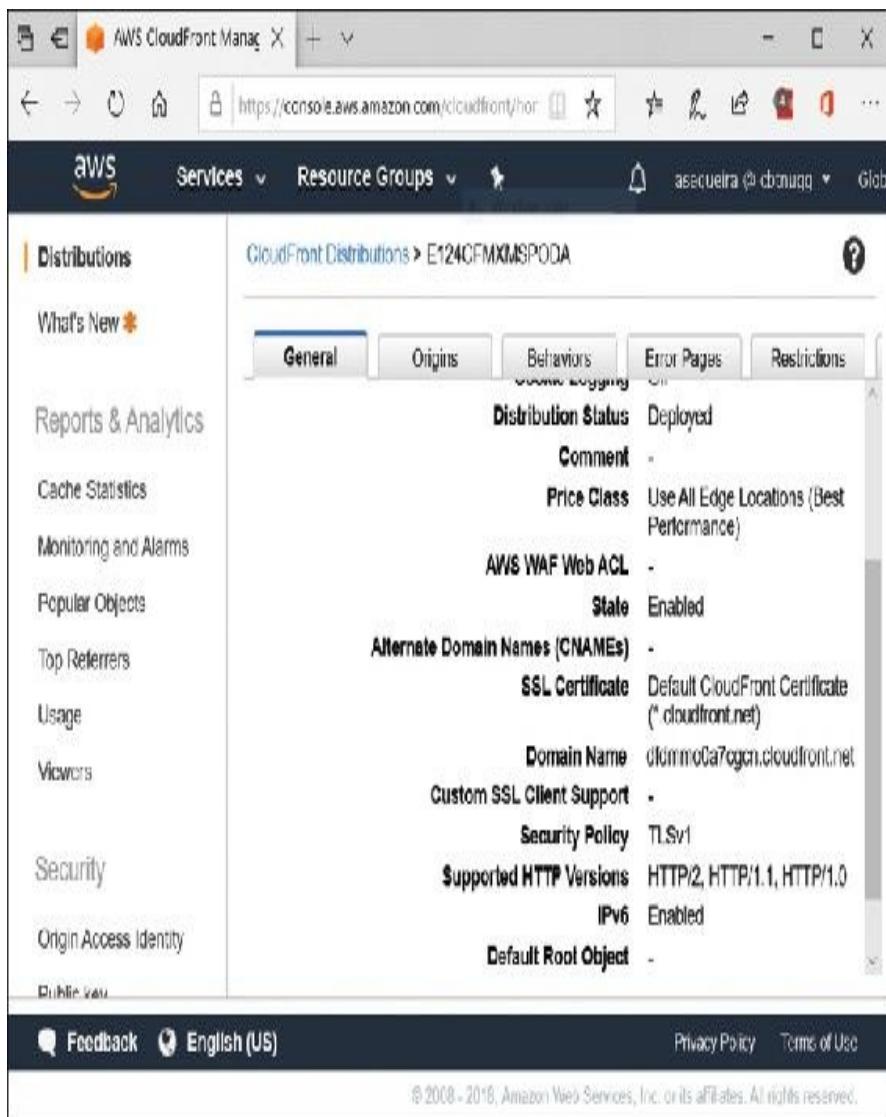


Figure 8-3 The CloudFront Distribution Properties

Step 13. In a web browser, enter your domain name and filename as a URL to test the CloudFront distribution. For example, my URL is http://dfdmmo0a7cgcn.cloudfront.net/DJI_0005.JPG. If your file is relatively large (like my JPG at 5 MB), you can see a performance difference with

your naked eye compared to retrieving from S3 directly because it is being served from a local edge location.

Lab Cleanup

Step 1. In the CloudFront Management Console, click the check box next to your distribution and click the **Disable** button. Click **Yes, Disable** and then click **Close**. Wait for the status of your distribution to return to Deployed. Once again, click the check box next to your distribution and choose **Delete**.

Step 2. In the Management Console, navigate to S3 and highlight your bucket by clicking an area to the right of it. Click **Delete Bucket** and then follow the instructions to delete your S3 bucket and its contents.

GREENGRASS

Related to caching is the functionality of the AWS service named Greengrass. This service falls under the Internet of Things (IoT) category. It allows IoT devices to collect and analyze data closer to the source on-premises and communicate this information with each other securely on the on-premises network. A common example of Greengrass usage is the development of serverless code (Lambda functions) written in AWS and then seamlessly delivered to IoT devices for local execution.

Fortunately, the IoT devices do not need to maintain constant connectivity to the cloud for Greengrass to function. Greengrass provides a local publisher/subscriber message

manager that can buffer messages to and from the cloud when connectivity is disrupted.

Greengrass consists of the following components:

- ■ Software distributions
 - ■ Greengrass core software
 - ■ Greengrass core SDK
- ■ Cloud service
 - ■ Greengrass API
- ■ Features
 - ■ Lambda runtime
 - ■ Shadows implementation (shadows represent your IoT devices and can be synced to the cloud)
 - ■ Message manager
 - ■ Group management
 - ■ Discovery service
 - ■ Over-the-air update agent
 - ■ Local resource access
 - ■ Machine learning inference

ROUTE 53

As you most likely know by now, Route 53 is the cloud-based DNS service of AWS. This service is covered here because caching in various forms plays a huge role in fast and efficient DNS communications. Although our main interest here is DNS time-to-live (TTL) values, it is worth covering the various Route 53 DNS concepts to ensure you are familiar with them:

Key Topic

- **Alias record:** This is a type of record you can create with Amazon Route 53 to direct traffic to AWS resources you control, such as CloudFront distributions or Amazon S3 buckets.

Note

Unlike a CNAME record, you can create an alias record at the top node of a DNS namespace, also known as the *zone apex*. For example, if you register the DNS name ajsnetworking.com, the zone apex is ajsnetworking.com. You cannot create a CNAME record for ajsnetworking.com, but you can create an alias record for the zone apex that routes traffic to www.ajsnetworking.com.

- **Authoritative name server:** This name server has definitive information about one part of the overall DNS system. A great example is an authoritative name server for the .com top-level domain (TLD). This name server knows the addresses of the name servers for every registered .com domain (that is a lot!); it returns this information when a user asks for it from a DNS resolver.
- **DNS resolver:** This DNS server acts as an intermediary between user requests (queries) and name servers; this is what you typically end up using when you send a DNS query from your web browser for a URL. Your system uses a DNS resolver managed by your ISP; it is often called a *recursive name server* because it sends requests to a sequence of authoritative name servers until it gets the response it needs.
- **Hosted zone:** This is a container object for the DNS records. It has the same name as the corresponding zone name; for example, my domain ajsnetworking.com has a hosted zone that contains the records for name resolution of my web server and mail server. It also contains information about my subdomains like private.ajsnetworking.com.
- **Private DNS:** This local version of Route 53 services directs traffic only for resources within your private VPCs in AWS.
- **DNS record:** This record enables you to determine how you want to direct traffic. For example, in my ajsnetworking.com domain, there is a DNS record for one of my web servers at 69.163.163.123.

- **Reusable delegation set:** You can use this set of four authoritative name servers with more than one hosted zone; this capability is useful when you are migrating a large number of zones to AWS.
- **Routing policy:** This setting for records controls query responses. Options are
 - **Simple routing policy:** This policy is used to direct traffic to a single resource that provides a given function—for example, a web server.
 - **Failover routing policy:** This policy can be used with active-passive failover.
 - **Geolocation routing policy:** This policy can be used to direct traffic to resources that are geographically close to users.
 - **Latency routing policy:** This policy is used to direct traffic to the lowest latency resource.
 - **Multivalue answer routing policy:** With this policy, DNS responds with up to eight healthy records selected at random.
 - **Weighted round robin:** This policy is used to direct traffic to multiple resources based on proportions that you can specify.
 - **Time to live (TTL):** This policy sets the amount of time (in seconds) that you want a DNS resolver to cache the values for a record before submitting another request to Route 53 for current values of that record.

For this discussion of caching, the TTL value is obviously of critical importance. You should cache DNS record information on DNS resolvers based on the amount of traffic that is in the DNS system and the speed with which you can return information that should not be changing all that often. When Route 53 returns a record with a TTL set to a fairly high value, your resolver will be able to service that client again quickly, as well as other clients looking for the same resource.

It is worth noting that although costs are not the primary concern of this chapter, setting higher TTL values does help

reduce Route 53 charges from AWS. The reason is that these charges are based on, in part, the number of requests that Route 53 must respond to. Clearly, you would need to set a small TTL if you are dealing with records that do, for whatever reason, change frequently.

You should also consider the impact of TTL for high availability and fault-tolerant configurations. If you want rapid failover to occur, short TTLs are a must. Consider a long TTL used with an Elastic Load Balancer. This would be a terrible idea. The ELB IP addresses are always subject to come and go as your architecture scales. If you are using a long TTL, the requester will get an IP back from the ELB service (via an ALIAS record typically) and cache it. If that ELB node/interface goes away or changes IP address, you could cause a perceived outage for that user (or users pointed to that previous address).

Lab: Creating a Hosted Domain and DNS Records in Route 53

In this lab you will practice with Route 53 by creating a sample hosted domain and DNS records inside the domain.

Step 1. Use the AWS Management Console to navigate to the **Route 53** link and select it.

Step 2. In the navigation links in the left column, choose **Hosted Zones**.

Step 3. Choose the **Create Hosted Zone** button, as shown in Figure 8-4.

Step 4. Enter the domain name **awsrocks.com** and make it a private hosted zone. Choose the default VPC of the

US West (Oregon) region. Then click **Create**.

Step 5. Click **Create Record Set**. Create a DNS record with the following parameters:

- ■ Name: **www**
- ■ Type: **A record for an IPv4 address**
- ■ TTL: **1 day**
- ■ Value: **10.10.10.100**
- ■ Routing Policy: **Simple**

Step 6. When you have completed the record details, click **Create**.

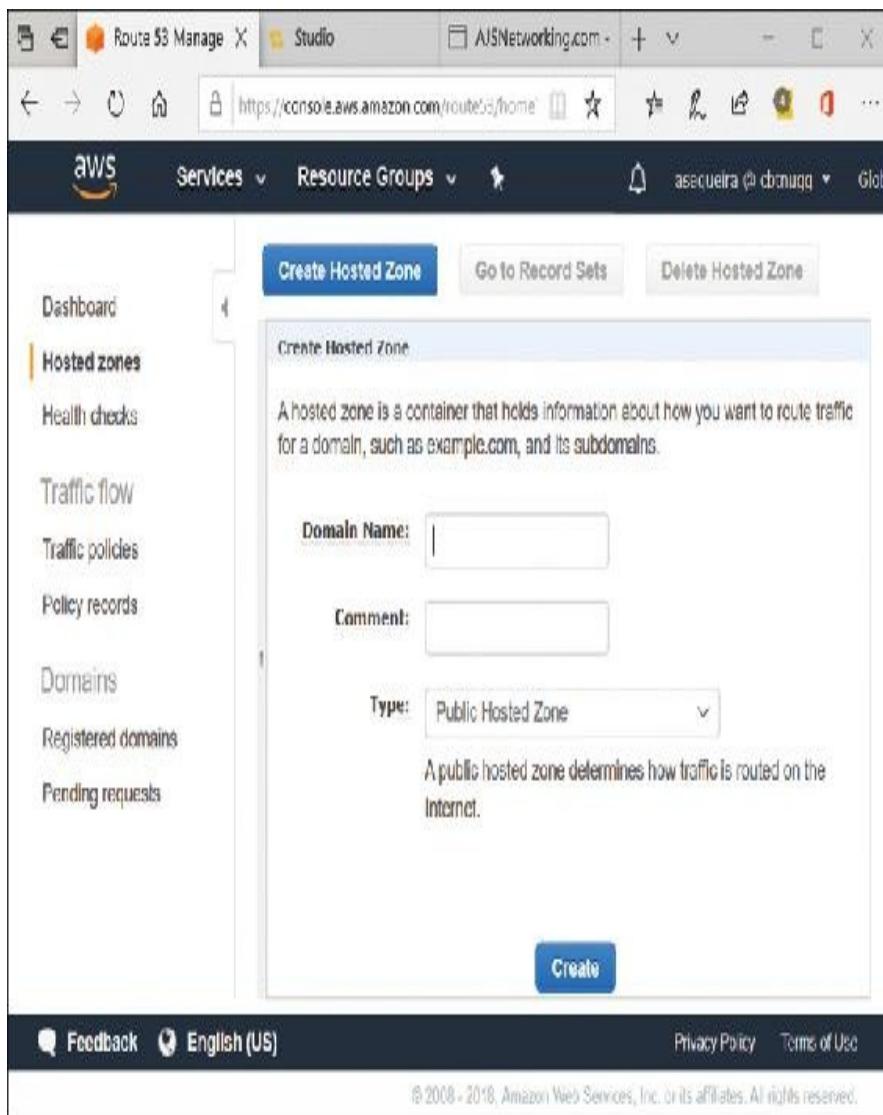


Figure 8-4 Creating a New Hosted Zone

Lab Cleanup

Step 1. Click the A record you created, choose **Delete Record Set**, and click **Confirm**.

Step 2. Click **Back to Hosted Zones**.

Step 3. Select the hosted zone you created in this lab and click **Delete Hosted Zone**.

Note

You cannot delete the hosted zone if it contains any records that you created in this lab.

EXAM PREPARATION TASKS

As mentioned in the section “How to Use This Book” in the Introduction, you have a couple of choices for exam preparation: the exercises here, [Chapter 16](#), “[Final Preparation](#),” and the exam simulation questions in the Pearson Test Prep practice test software.

REVIEW ALL KEY TOPICS

Review the most important topics in this chapter, noted with the Key Topics icon in the margin of the page. [Table 8-2](#) lists a reference of these key topics and the page numbers on which each is found.



Table 8-2 Key Topics for [Chapter 8](#)

Key Topic Element	Description	Page Number
Concept	ElastiCache engine support	142
List	DynamoDB use cases	146

Steps	Configuration of CloudFront	147
List	Route 53 terminology	150

COMPLETE TABLES AND LISTS FROM MEMORY

There are no memory tables in this chapter.

DEFINE KEY TERMS

Define the following key terms from this chapter and check your answers in the glossary:

Redis

memcached

DAX

Edge location

Distribution

Greengrass

TTL

Q&A

The answers to these questions appear in Appendix A. For more practice with exam format questions, use the Pearson Test Prep practice test software.

1. What are the two caching engines supported by ElastiCache?
2. DAX is made up of nodes in what type of structure?
3. What type of data center does CloudFront use to store cached information?
4. What are typical code pieces that Greengrass will synch to local IoT devices?
5. What does a Hosted Zone in Route 53 consist of?

Chapter 9. Designing for Elasticity

This chapter covers the following subjects:

- ■ **Elastic Load Balancing:** The capability of your AWS infrastructure to dynamically share traffic among multiple resources (such as EC2 instances) hinges on the ELB feature. This section discusses Elastic Load Balancing in detail.
- ■ **Auto Scaling:** To provide elasticity, you can have the number of EC2 instances increase or decrease based on several factors. This section describes Auto Scaling in great detail.

We discussed elasticity briefly when we presented the various advantages that cloud technologies provide IT solutions in Chapter 1, “The Fundamentals of AWS.” Remember, *elasticity* refers to the capability of your resources to scale up and down as well as in and out when resource requirements change due to changes in demand.

“DO I KNOW THIS ALREADY?” QUIZ

The “Do I Know This Already?” quiz allows you to assess whether you should read the entire chapter. Table 9-1 lists the major headings in this chapter and the “Do I Know This Already?” quiz questions covering the material in those headings so you can assess your knowledge of these specific areas. The answers to the “Do I Know This Already?” quiz appear in Appendix A.

Table 9-1 “Do I Know This Already?” Foundation Topics
Section-to-Question Mapping

Foundation Topics Section	Question
Elastic Load Balancing	1–2
Auto Scaling	3–4

Caution

The goal of self-assessment is to gauge your mastery of the topics in this chapter. If you do not know the answer to a question or are only partially sure of the answer, you should mark that question as wrong for purposes of the self-assessment. Giving yourself credit for an answer you correctly guess skews your self-assessment results and might provide you with a false sense of security.

- 1.** Which is not a common service used with ELB?
 - a.** EC2
 - b.** CloudWatch
 - c.** EBS
 - d.** Route 53
- 2.** Which of the following is not a valid type of Elastic Load Balancer in AWS?
 - a.** Classic
 - b.** Service-based
 - c.** Network
 - d.** Application
- 3.** How many Auto Scaling policies can you set against an

AWS resource?

- a. 1
- b. 10
- c. 50
- d. 100

4. What is the cooldown period used with Auto Scaling?

- a. It is the mandatory time that a service must run before sampling can begin.
- b. It is the mandatory sample time of metrics for policy-based Auto Scaling.
- c. It is the mandatory time that Auto Scaling must wait after a service reboots before scale-out operations.
- d. It is the mandatory time that Auto Scaling must wait before it takes additional scaling actions.

FOUNDATION TOPICS

ELASTIC LOAD BALANCING

Elastic Load Balancing (ELB) distributes incoming application or network traffic across multiple targets. These targets are AWS resources such as Amazon EC2 instances, containers, and IP addresses. To foster high availability, ensure the resources are in multiple Availability Zones.

Elastic Load Balancing scales your load balancer as traffic to your application changes over time and can scale to a majority of workloads automatically. You can add and remove compute resources from your load balancer as your needs change,

without disrupting the overall flow of requests to your applications.

You can configure health checks, which are used to monitor the health of the computing resources so that the load balancer can send requests only to the healthy ones. You can also offload the work of SSL/TLS encryption and decryption to your load balancer so that your compute resources can focus on their main work.

Elastic Load Balancing supports three types of load balancers: Application Load Balancers, Network Load Balancers, and Classic Load Balancers. You can select a load balancer based on your application needs. Figure 9-1 shows the configuration of a Network Load Balancer in AWS.

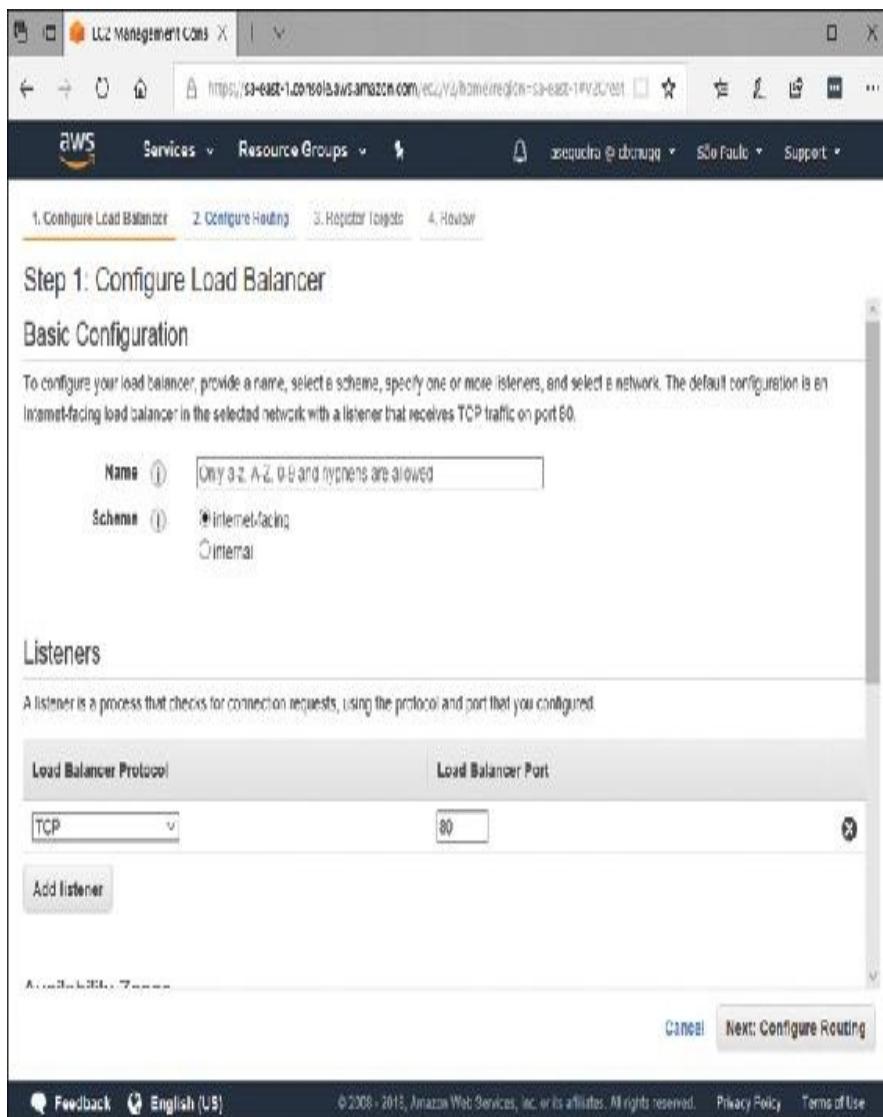


Figure 9-1 Configuring a Network Load Balancer in AWS

Although you can monitor and configure your Elastic Load Balancer with the traditional options of AWS (such as the Management Console or SDKs), you can also use the Query API. This interface provides low-level API actions that you call using HTTPS requests.

Elastic Load Balancing works with the following services to improve the availability and scalability of your applications:

Key Topic

- **■ EC2:** These virtual servers run your applications in the cloud. You can configure your load balancer to route traffic to your EC2 instances.
- **■ ECS:** This service enables you to run, stop, and manage Docker containers on a cluster of EC2 instances. You can configure your load balancer to route traffic to your containers.
- **■ Auto Scaling:** This service ensures that you are running your desired number of instances, even if an instance fails, and enables you to automatically increase or decrease the number of instances as the demand on your instances changes. If you enable Auto Scaling with Elastic Load Balancing, instances that are launched by Auto Scaling are automatically registered with the load balancer, and instances that are terminated by Auto Scaling are automatically deregistered from the load balancer.
- **■ CloudWatch:** This service enables you to monitor your load balancer and take action as needed.
- **■ Route 53:** This service provides a reliable and cost-effective way to route visitors to websites by translating domain names (such as www.ajsnetworking.com) into the numeric IP addresses (such as 69.163.163.123) that computers use to connect to each other. AWS assigns URLs to your resources, such as load balancers. However, you might want a URL that is easy for users to remember. For example, you can map your domain name to a load balancer.

AUTO SCALING

Auto Scaling enables you to quickly discover the scalable AWS resources that are part of your application and configure dynamic scaling in a matter of minutes. The Auto Scaling console provides a single user interface to use the automatic scaling features of multiple AWS services. It also offers recommendations to configure scaling for the scalable resources in your application.

Use Auto Scaling to automatically scale the following

resources that support your application:

Key Topic

- ■ EC2 Auto Scaling groups
- ■ Aurora DB clusters
- ■ DynamoDB global secondary indexes
- ■ DynamoDB tables
- ■ ECS services
- ■ Spot Fleet requests

With Auto Scaling, you create a scaling plan with a set of instructions used to configure dynamic scaling for the scalable resources in your application. Auto Scaling creates target tracking scaling policies for the scalable resources in your scaling plan. Target tracking scaling policies adjust the capacity of your scalable resource as required to maintain resource utilization at the target value that you specified.

You can create one scaling plan per application source (an AWS CloudFormation stack or a set of tags). You can add each scalable resource to one scaling plan. If you have already configured scaling policies for a scalable resource in your application, Auto Scaling keeps the existing scaling policies instead of creating additional scaling policies for the resource.

Auto Scaling involves the creation of a Launch Configuration and an Auto Scaling group. [Figure 9-2](#) shows the configuration of Auto Scaling in AWS.

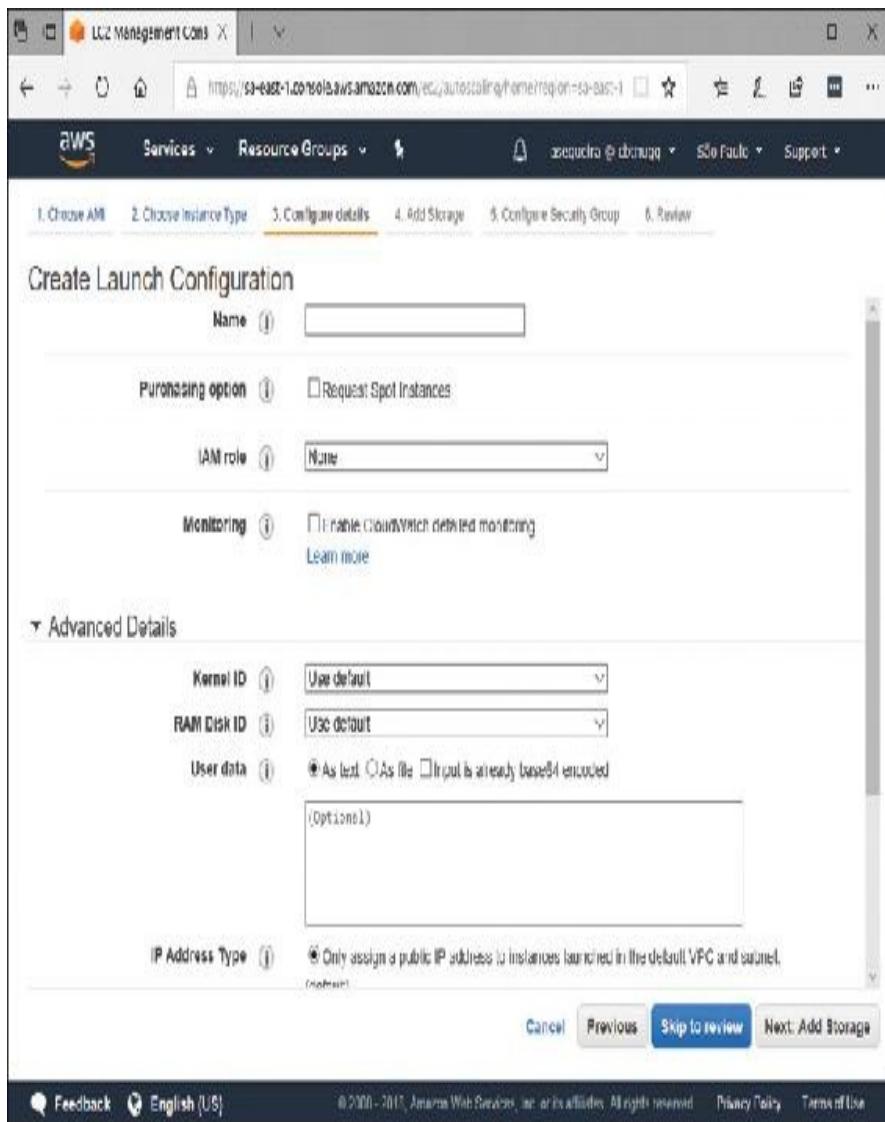


Figure 9-2 Configuring Auto Scaling in AWS

Target Tracking Scaling Policies

With target tracking scaling policies, you select a predefined metric or configure a customized metric and set a target value. Application Auto Scaling creates and manages the CloudWatch alarms that trigger the scaling policy and calculates the scaling adjustment based on the metric and the target value. The scaling policy adds or removes capacity as required to keep the metric at, or close to, the specified target.

value. In addition to keeping the metric close to the target value, a target tracking scaling policy also adjusts to changes in the metric due to a changing load pattern and minimizes changes to the capacity of the scalable target.

When specifying a customized metric, be aware that not all metrics work for target tracking. The metric must be a valid utilization metric and describe how busy a scalable target is. The metric value must increase or decrease proportionally to the capacity of the scalable target so that the metric data can be used to proportionally scale the scalable target.

You can have multiple target tracking scaling policies for a scalable target, provided that each of them uses a different metric. Application Auto Scaling scales based on the policy that provides the largest capacity for both scale-in and scale-out. This provides greater flexibility to cover multiple scenarios and ensures that there is always enough capacity to process your application workloads.

Note

When discussing scaling the resources of a service, we are scaling those resources horizontally (out and in with elasticity), while the service made up of those resources is being scaled up and down (vertically because the single service is getting bigger or smaller). A single service scales both up and down and out and in, depending on the context.

You can also optionally disable the scale-in portion of a target tracking scaling policy. This feature provides the flexibility to use a different method for scale-in than you use for scale-out.

Keep the following in mind for Auto Scaling:

- You cannot create target tracking scaling policies for Amazon EMR clusters or AppStream 2.0 fleets.

- ■ You can create 50 scaling policies per scalable target. This includes both step scaling policies and target tracking policies.
- ■ A target tracking scaling policy assumes that it should perform scale-out when the specified metric is above the target value. You cannot use a target tracking scaling policy to scale out when the specified metric is below the target value.
- ■ A target tracking scaling policy does not perform scaling when the specified metric has insufficient data. It does not perform scale-in because it does not interpret insufficient data as low utilization. To scale in when a metric has insufficient data, create a step scaling policy and have an alarm invoke the scaling policy when it changes to the INSUFFICIENT_DATA state.
- ■ You may see gaps between the target value and the actual metric data points. The reason is that Application Auto Scaling always acts conservatively by rounding up or down when it determines how much capacity to add or remove. This prevents it from adding insufficient capacity or removing too much capacity. However, for a scalable target with small capacity, the actual metric data points might seem far from the target value. For a scalable target with larger capacity, adding or removing capacity causes less of a gap between the target value and the actual metric data points.
- ■ We recommend that you scale based on metrics with a 1-minute frequency because that ensures a faster response to utilization changes. Scaling on metrics with a 5-minute frequency can result in slower response time and scaling on stale metric data.
- ■ To ensure application availability, Application Auto Scaling scales out proportionally to the metric as fast as it can but scales in more gradually.
- ■ Do not edit or delete the CloudWatch alarms that Application Auto Scaling manages for a target tracking scaling policy. Application Auto Scaling deletes the alarms automatically when you delete the Auto Scaling policy.

The Cooldown Period

The scale-out cooldown period is the amount of time, in seconds, after a scale-out activity completes before another scale-out activity can start. While this cooldown period is in

effect, the capacity that has been added by the previous scale-out event that initiated the cooldown is calculated as part of the desired capacity for the next scale-out event. The intention is to continuously scale out.

The scale-in cooldown period is the amount of time, in seconds, after a scale-in activity completes before another scale-in activity can start. This cooldown period is used to block subsequent scale-in events until it has expired. The intention is to scale in conservatively to protect your application's availability. However, if another alarm triggers a scale-out policy during the cooldown period after a scale-in event, Application Auto Scaling scales out your scalable target immediately.

EXAM PREPARATION TASKS

As mentioned in the section “How to Use This Book” in the Introduction, you have a couple of choices for exam preparation: the exercises here, [Chapter 16, “Final Preparation,”](#) and the exam simulation questions in the Pearson Test Prep practice test software.

REVIEW ALL KEY TOPICS

Review the most important topics in this chapter, noted with the Key Topics icon in the margin of the page. [Table 9-2](#) lists a reference of these key topics and the page numbers on which each is found.

Table 9-2 Key Topics for [Chapter 9](#)

Key Topic

Key Topic Element	Description	Page Number
List	Resources you can use with Elastic Load Balancing	160
List	Resources you can use with Auto Scaling	160

COMPLETE TABLES AND LISTS FROM MEMORY

There are no memory tables in this chapter.

DEFINE KEY TERMS

Define the following key term from this chapter and check your answer in the glossary:

Cooldown period

Q&A

The answers to these questions appear in Appendix A. For more practice with exam format questions, use the Pearson Test Prep practice test software.

1. What are the three types of load balancers in AWS?

2. Do you need to delete the individual CloudWatch alarms that you use for Auto Scaling when you delete the Auto Scaling policy ?

Part III

Domain 3: Specify Secure Applications and Architectures

Chapter 10. Securing Application Tiers

This chapter covers the following subjects:

- **Using IAM:** Identity and Access Management in AWS is the foundation of proper security. This section ensures that you are well versed in how to use the various components it contains.
- **Securing the OS and Applications:** This section examines two critical components of effective security in AWS: security groups and network ACLs. It also discusses AWS Systems Manager and its capability to assist you in the important task of patching operating systems running in your infrastructure.

Security is a huge concern for any IT infrastructure but is especially a topic of concern in a cloud-based implementation. The reason is the many perceived risk points with this new paradigm. This chapter ensures you understand key elements of AWS security, including IAM, security groups, and network ACLs.

“DO I KNOW THIS ALREADY?” QUIZ

The “Do I Know This Already?” quiz allows you to assess whether you should read the entire chapter. Table 10-1 lists the major headings in this chapter and the “Do I Know This Already?” quiz questions covering the material in those headings so you can assess your knowledge of these specific areas. The answers to the “Do I Know This Already?” quiz appear in [Appendix A](#).

Table 10-1 “Do I Know This Already?” Foundation Topics
Section-to-Question Mapping

Foundation Topics Section	Questions
Using IAM	1–4
Securing the OS and Applications	5–6

Caution

The goal of self-assessment is to gauge your mastery of the topics in this chapter. If you do not know the answer to a question or are only partially sure of the answer, you should mark that question as wrong for purposes of the self-assessment. Giving yourself credit for an answer you correctly guess skews your self-assessment results and might provide you with a false sense of security.

- 1.** What type of account has full administrative privileges associated with it and is accessed by email address?
 - a.** Console access account
 - b.** Root user account
 - c.** API control account
 - d.** Role account

- 2.** You have configured your account for administration of AWS; you require a code to be entered that is delivered to your cell phone in addition to your AWS password. What is this approach called?
 - a.** MFA

- b. SAML**
 - c. UTP**
 - d. SAM**
- 3.** What IAM component seeks to improve scalability and administrative ease in configuration?
 - a. Roles**
 - b. Users**
 - c. Security Groups**
 - d. Network ACLs**
- 4.** You need users that have already authenticated through your Active Directory to be granted access to your AWS resources for several administrative tasks. What is this type of access called?
 - a. Proxied**
 - b. Forwarded**
 - c. Deferred**
 - d. Federated**
- 5.** Which of these characteristics is not true of network ACLs?
 - a. They are assigned to subnets.**
 - b. They consist of permit and deny entries.**
 - c. They are stateful in their operation.**
 - d. They consist of entries processed in order.**
- 6.** What are security groups applied to in AWS?
 - a. User accounts**

- b.** EC2 instances
- c.** Subnets
- d.** Network interfaces

FOUNDATION TOPICS

USING IAM

AWS Identity and Access Management (IAM) permits you to securely control access to AWS resources. You use IAM to control who is authenticated and authorized to use resources.

Figure 10-1 shows the management console interface of IAM.

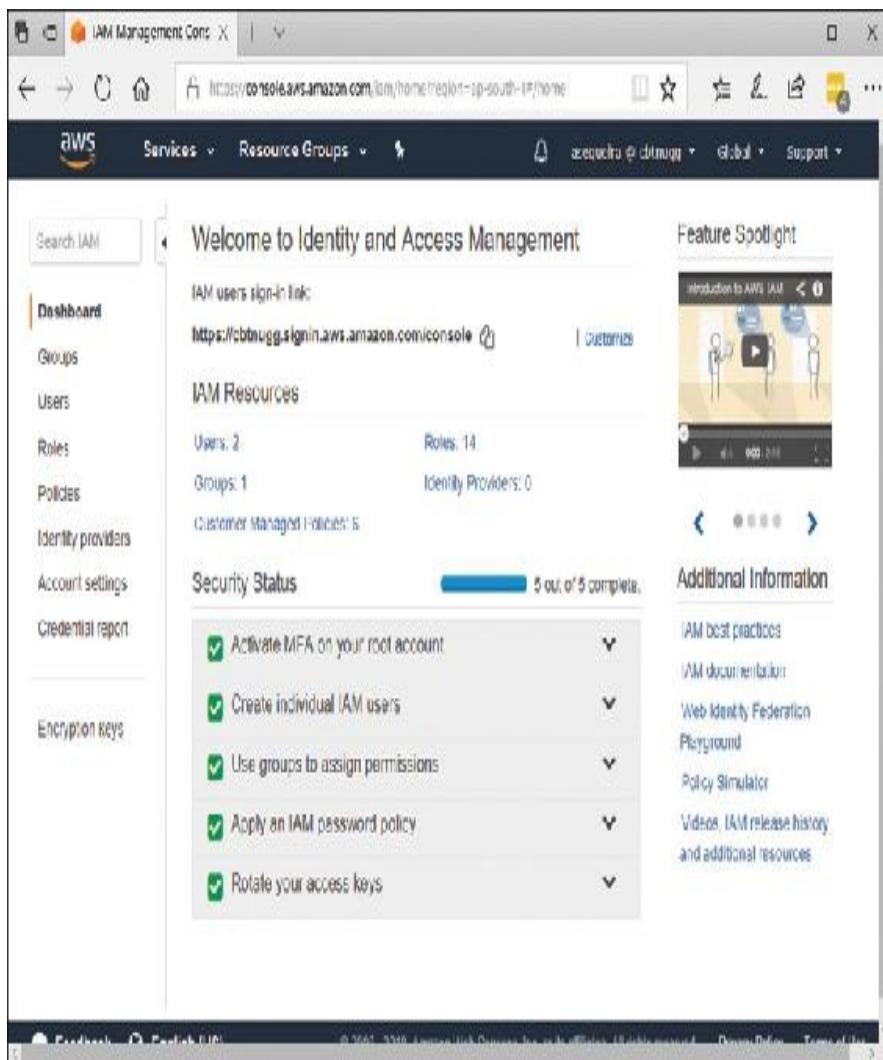


Figure 10-1 Identity and Access Management (IAM) in AWS

When you first create your AWS account (typically a free tier account), you begin with a single sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account root user. You access this account by signing in with the email address and password that you used to create the account.

You should not use the root user for your everyday tasks, even your everyday administrative tasks. You should log in as the

root user only to create your first IAM user. Then you should securely lock away the root user credentials, add multifactor authentication (MFA) to this account, and use the root user credentials only to perform the few administrative tasks where this all-powerful account is required.

You should familiarize yourself with these features of IAM in AWS:



- ■ **Shared access to your AWS account:** You can grant other people permission to administer and use resources in your AWS account without having to share your password or access key. You typically do this using roles.
- ■ **Granular permissions:** You can grant different permissions to different people for different resources.
- ■ **Secure access to AWS resources for applications that run on Amazon EC2:** You can use IAM features to securely provide credentials for applications that run on EC2 instances. These credentials provide permissions for your application to access other AWS resources. Once again, this is often accomplished with the roles component in IAM.
- ■ **Multifactor authentication (MFA):** You can add two-factor authentication to your root account and to individual IAM users for extra security. With MFA, you or your users must provide not only a password or access key to work with your account but also a code from a specially configured device. The most common approach is to use a cell phone to provide the second authentication component.
- ■ **Identity federation:** You can allow users who already have passwords elsewhere to get temporary access to your AWS account.
- ■ **Identity information for assurance:** If you use AWS CloudTrail, you receive log records that include information about those who made requests for resources in your account. That information is based on IAM identities.
- ■ **PCI DSS Compliance:** IAM supports the processing, storage, and transmission of credit card data by a merchant or service provider. It has

been validated as being compliant with the Payment Card Industry (PCI) Data Security Standard (DSS).

- **■ Integration with many AWS services:** IAM can be used across AWS services and resources seamlessly.
- **■ Eventually consistent:** IAM, like many other AWS services, is eventually consistent. IAM achieves high availability by replicating data across multiple servers within Amazon's data centers around the world. If a request to change some data is successful, the change is committed and safely stored. However, the change must be replicated across IAM, which can take some time. Such changes include creating or updating users, groups, roles, or policies. Amazon recommends that you do not include such IAM changes in the critical, high-availability code paths of your application. Instead, make IAM changes in a separate initialization or setup routine that you run less frequently. Also, be sure to verify that the changes have been propagated before production workflows depend on them.
- **■ Free use:** IAM and AWS Security Token Service (AWS STS) are features of your AWS account offered at no additional charge.

You can work with AWS Identity and Access Management in any of the following ways:

- **■ AWS Management Console**
- **■ AWS command-line tools**
- **■ AWS SDKs**
- **■ IAM HTTPS API**

IAM Identities

What most people think of immediately when they hear *IAM* and *AWS* is the main identities of users, groups, and roles.

An IAM user is an entity that you create in AWS. The IAM user represents the person or service that uses the IAM user to interact with AWS. A primary use for IAM users is to give one of your cloud engineers the ability to sign in to the AWS Management Console for interactive tasks and to make

programmatic requests to AWS services using the API or CLI.

A user in AWS consists of a name, a password to sign in to the AWS Management Console, and up to two access keys that can be used with the API or CLI. When you create an IAM user, you should grant it permissions by making it a member of a group that has appropriate permission policies attached. Although you could attach these policies directly to the user account, this approach is usually terrible for scalability in your security design. In fact, most seasoned AWS administrators use groups even if these groups contain only a single user account.

Note

You can clone the permissions of an existing IAM user, which automatically makes the new user a member of the same groups and attaches all the same policies.

An IAM group is a collection of IAM users. You can use groups to specify permissions for a collection of users, which can make those permissions easier to manage for those users. Figure 10-2 shows an example of a group in AWS.

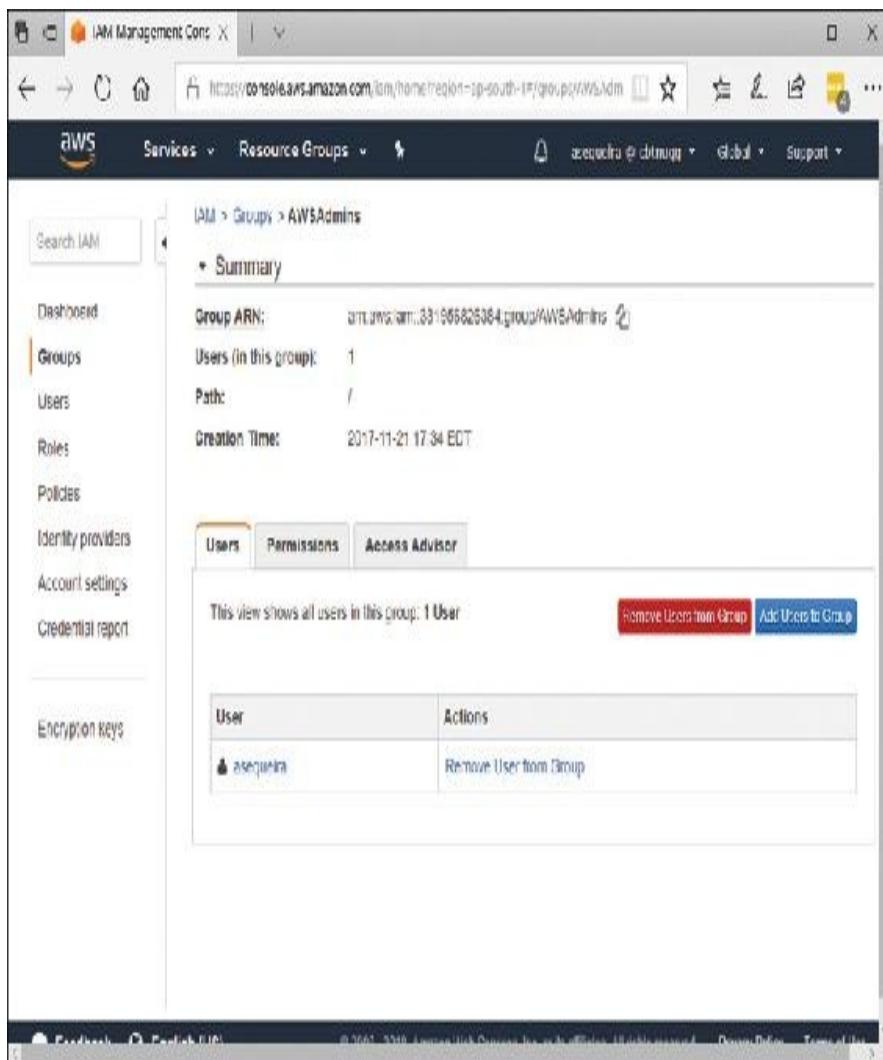


Figure 10-2 A Group in IAM in AWS

Note

A group is not truly an identity because it cannot be identified as a principal in a resource-based or trust policy. It is only a way to attach policies to multiple users at one time.

An IAM role is similar to a user, in that it is an identity with permission policies that determine what the identity can and cannot do in AWS. However, a role does not have any credentials associated with it. Instead of being uniquely

associated with one person, a role is intended to be assumable by anyone who needs it. An IAM user can assume a role to temporarily take on different permissions for a specific task. A role can be assigned to a federated user who signs in by using an external identity provider instead of IAM. AWS uses details passed by the identity provider to determine which role is mapped to the federated user.

Another important component of IAM is policies. In fact, you actually manage access in AWS by creating policies and attaching them to IAM identities like groups and roles. You can also attach policies to AWS resources themselves. Most policies are stored in AWS as JSON documents. In AWS, there are several different policy types as follows:

- **Identity-based policies:** These policies are attached to users, groups, or roles.
- **Resource-based policies:** These policies are attached to resources in AWS such as S3 buckets.
- **Organization SCPs:** These policies allow the creation of a Service Control Policy (SCP); this permits you to define a permissions boundary to an AWS organization or organizational unit (OU).
- **Access control lists (ACLs):** These lists control what principals (users/roles/groups) can access a resource; this is the only policy type that does not use a JSON policy document structure.

The JSON policy document structure consists of an optional policy-wide section for information at the top of the document and then one or more individual statements about permissions. Here is an example:

```
{ "Version": "2019-1-19",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "s3>ListBucket",
      "Resource": "arn:aws:s3::<bucket-name>"}
```

```
    "Resource": "arn:aws:s3:::asequeirabucket123"
}
}
```

This example permits the implied principal to list a single S3 bucket named asegueirabucket123.

SECURING THE OS AND APPLICATIONS

You should understand many security components beyond IAM to effectively secure your AWS infrastructure. Two key components are security groups and network ACLs.

Unfortunately, these security mechanisms are often poorly understood and often confused with each other. The sections that follow provide details about each of these security structures you should know.

Security Groups

Here are key aspects of security groups that you should be aware of:



- Security groups apply to network interfaces to control access to EC2 instances (and potentially other AWS resources).
- Security groups are stateful in their operation; for example, a traffic flow explicitly permitted outbound will trigger a dynamic inbound permission of the return flow for that communication requirement.
- Security groups are built by your PERMIT entries. With security groups, you do not create DENY entries.
- You can reference other security groups from your VPC in your security group; you can also reference security groups in other VPCs that you have peered with.

Figure 10-3 shows an example of a security group in AWS.

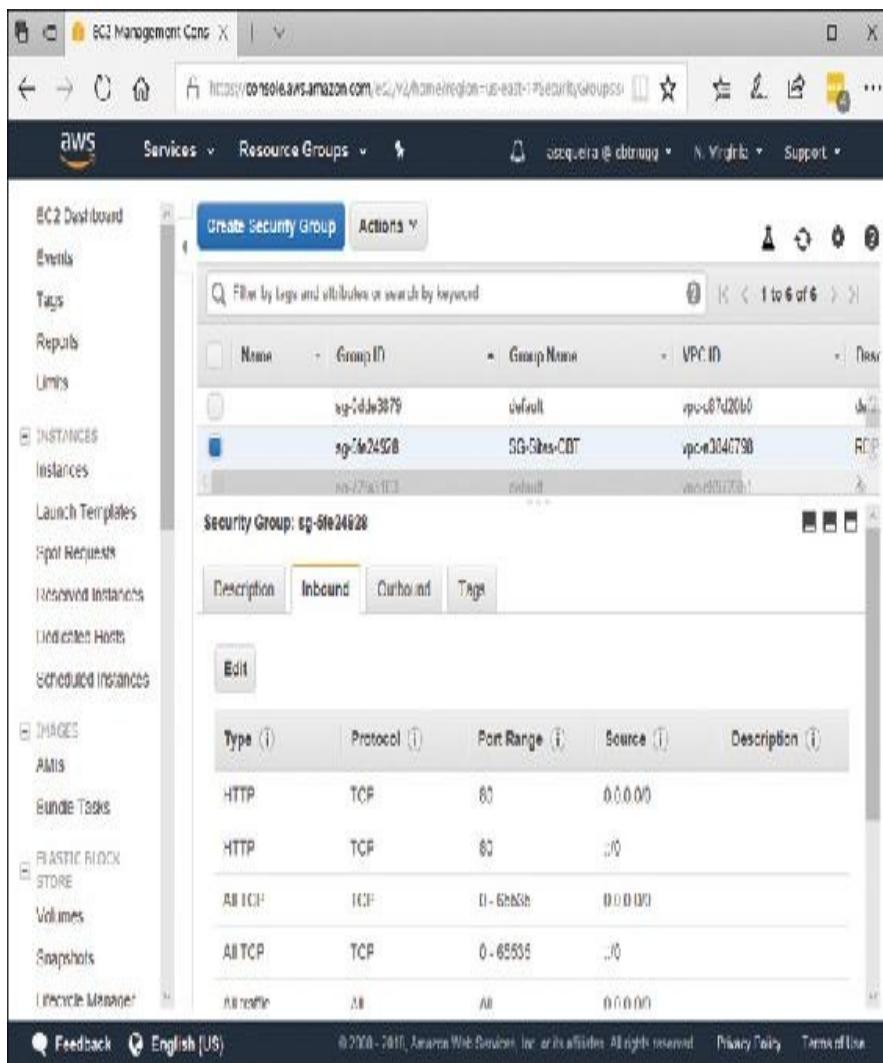


Figure 10-3 A Security Group in AWS

Network ACLs

Network ACLs are overall less functional for use in AWS because they protect traffic entering or exiting subnets of AWS. I consider them less functional because subnets are not the typical security boundary you work with in AWS. Network ACLs are the primary tool you would use to deny specific IP addresses from accessing resources because this cannot be accomplished with security groups. Consider these important points regarding network ACLs:

Key Topic

- Network ACLs apply to subnets to control traffic in and out of these subnets.
- Network ACLs are stateless in their operation; you must explicitly permit return flows based on flows permitted in the opposite direction.
- You can use PERMIT and DENY entries with your network ACLs.
- Network ACL entries are processed in order, and order is critically important.

Remember, a key to securing your cloud-based infrastructure (and even traditional infrastructures) is to keep your operating systems patched and updated. AWS provides the Systems Manager service, which offers [Patch Manager](#) to assist in this regard.

Systems Manager Patch Manager

AWS Systems Manager Patch Manager automates the process of patching managed instances with security-related updates. For Linux-based instances, you can also install patches for nonsecurity updates. You can patch fleets of Amazon EC2 instances or your on-premises servers and virtual machines by operating system type. This includes supported versions of Windows, Ubuntu Server, Red Hat Enterprise Linux (RHEL), SUSE Linux Enterprise Server (SLES), Amazon Linux, and Amazon Linux 2. You can scan instances to see only a report of missing patches, or you can scan and automatically install all missing patches.

Patch Manager uses patch baselines, which include rules for auto-approving patches within days of their release, as well as a list of approved and rejected patches. You can install patches

on a regular basis by scheduling patching to run as a Systems Manager Maintenance Window task. You can also install patches individually or to large groups of instances by using Amazon EC2 tags.

Patch Manager integrates with IAM, CloudTrail, and CloudWatch events to provide a secure patching experience that includes event notifications and the ability to audit usage. Figure 10-4 shows the Patch Manager interface in AWS.

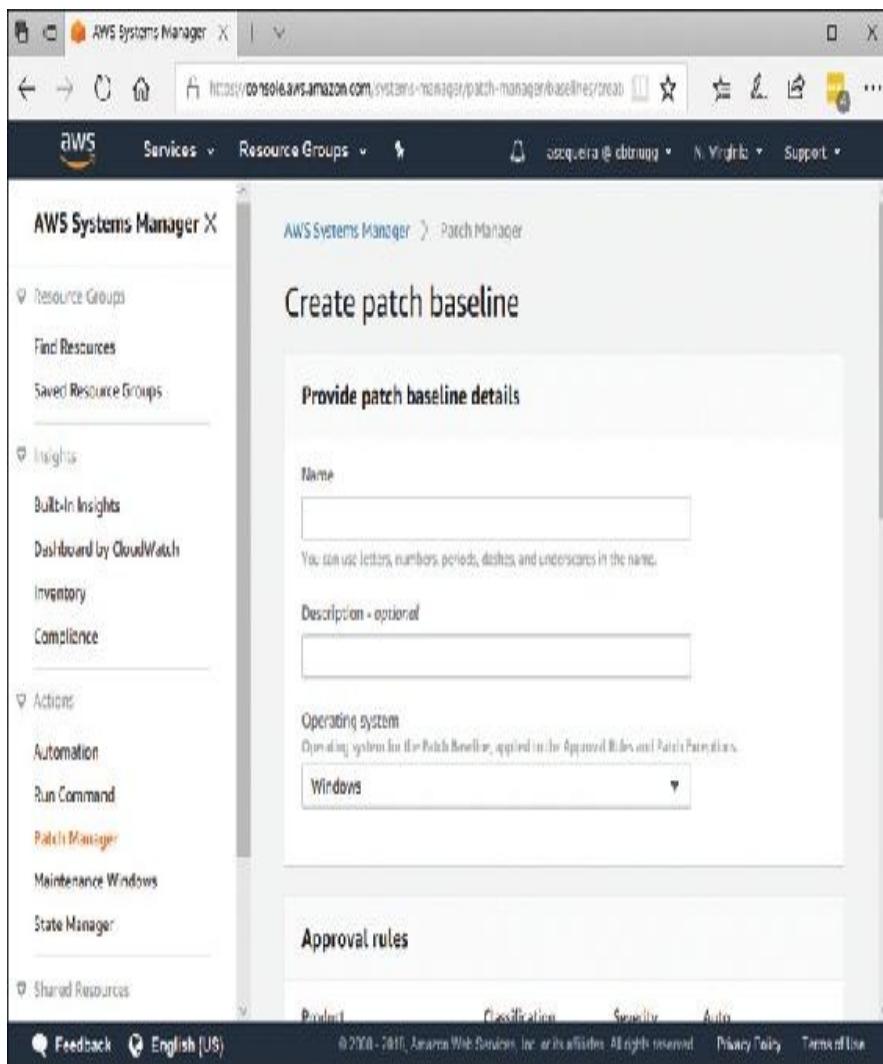


Figure 10-4 Patch Manager of the AWS Systems Manager

EXAM PREPARATION TASKS

As mentioned in the section “How to Use This Book” in the Introduction, you have a couple of choices for exam preparation: the exercises here, [Chapter 16, “Final Preparation,”](#) and the exam simulation questions in the Pearson Test Prep practice test software.

REVIEW ALL KEY TOPICS

Review the most important topics in this chapter, noted with the Key Topics icon in the margin of the page. [Table 10-2](#) lists a reference of these key topics and the page numbers on which each is found.



Table 10-2 Key Topics for Chapter 10

Key Topic Element	Description	Page Number
List	Key features of IAM	170
List	Security groups	174
List	Network ACLs	175

COMPLETE TABLES AND LISTS FROM MEMORY

There are no memory tables in this chapter.

DEFINE KEY TERMS

Define the following key terms from this chapter and check your answers in the glossary:

Root user

Users

Groups

Roles

Policy

Security group

Network ACL

Patch Manager

Q&A

The answers to these questions appear in Appendix A. For more practice with exam format questions, use the Pearson Test Prep practice test software.

- 1.** What are the three main identity components in AWS IAM?

- 2.** What AWS tool can assist with patching operating systems with important security updates?

Chapter 11. Securing Data

This chapter covers the following subjects:

- **Resource Access Authorization:** This section discusses the process of providing the correct authorization to identities that need to interact with AWS resources.
- **Storing and Managing Encryption Keys in the Cloud:** This section describes the power and flexibility provided by the encryption key management capabilities of AWS.
- **Protecting Data at Rest:** Encryption is one of the keys to securing data at rest. This section describes the three approaches to encrypting your data at rest. It also provides examples of where you might encrypt data at rest in AWS.
- **Decommissioning Data Securely:** This section discusses how AWS can ensure that you dispose properly of data that is cloud-based.
- **Protecting Data in Transit:** This section describes the mechanism you have available in AWS to ensure data is protected while it is being transferred.

This chapter is the second one that has a security focus. In this chapter, you learn about securing data at rest as well as securing data that is in transit. This chapter also discusses other aspects of security for the lifecycle that data transitions through.

“DO I KNOW THIS ALREADY?” QUIZ

The “Do I Know This Already?” quiz allows you to assess whether you should read the entire chapter. Table 11-1 lists the major headings in this chapter and the “Do I Know This Already?” quiz questions covering the material in those headings so you can assess your knowledge of these specific

areas. The answers to the “Do I Know This Already?” quiz appear in Appendix A.

Table 11-1 “Do I Know This Already?” Foundation Topics
Section-to-Question Mapping

Foundation Topics Section	Questions
Resource Access Authorization	1
Storing and Managing Encryption Keys in the Cloud	2
Protecting Data at Rest	3
Decommissioning Data Securely	4
Protecting Data in Transit	5

Caution

The goal of self-assessment is to gauge your mastery of the topics in this chapter. If you do not know the answer to a question or are only partially sure of the answer, you should mark that question as wrong for purposes of the self-assessment. Giving yourself credit for an answer you correctly guess skews your self-assessment results and might provide you with a false sense of security.

- 1.** What acts as a firewall protecting access to EC2 instances against unauthorized access and administration?

 - a.** Security group
 - b.** Role
 - c.** Federation
 - d.** Storage Gateway
- 2.** What is the AWS service for management of all encryption needs in the cloud?

 - a.** Greenfield
 - b.** Kinesis
 - c.** KMS
 - d.** Cognito
- 3.** What are your options when it comes to encrypting content at rest in S3? (Select all that apply.)

 - a.** You manage everything involving encryption.
 - b.** AWS manages everything except the management functions for keys.
 - c.** There is no management required because you use a keyless solution.
 - d.** AWS manages everything involving encryption.
- 4.** What standards can you have AWS meet for you when it comes to data decommissioning?

 - a.** NIST
 - b.** IEEE

- c. OSHA
 - d. IC2
5. What protocol is a key to protecting data in transit with AWS?
- a. HTTP
 - b. DES
 - c. ICMPS
 - d. HTTPS

FOUNDATION TOPICS

RESOURCE ACCESS AUTHORIZATION

So that you can fully understand how resource access authorization can function in AWS, let's use the example of EC2 access. EC2 is a common component used in AWS architectures, and it provides a common example that also applies to many other resources.

Remember that your security credentials identify you to services in AWS and grant you potentially unlimited use of your AWS resources if the policy permits. You can use features of EC2 and Identity and Access Management (IAM) to



- Allow other users, services, and applications to use your EC2 resources without sharing your security credentials.
- Control how other users use resources in your AWS account.

- ■ Use security groups to control access to your EC2 instances.
- ■ Allow full use or limited use of your EC2 resources.

As you know from [Chapter 10, “Securing Application Tiers,”](#) a security group acts as a firewall that controls the traffic allowed to reach one or more instances. When you launch an instance, you assign it one or more security groups. You add rules to each security group that control traffic for the instance. You can modify the rules for a security group at any time; the new rules are automatically applied to all instances to which the security group is assigned.

Your organization might have multiple AWS accounts. EC2 enables you to specify additional AWS accounts that can use your AMIs and EBS snapshots. All users in the AWS account that you have specified can use the AMI or snapshot. Each AMI has a `LaunchPermission` attribute that controls which AWS accounts can access the AMI. Each Amazon EBS snapshot has a `createVolumePermission` attribute that controls which AWS accounts can use the snapshot.

IAM enables you to do the following:

- ■ Create users and groups under your AWS account
- ■ Assign unique security credentials to each user under your AWS account
- ■ Control each user’s permissions to perform tasks using AWS resources
- ■ Allow the users in another AWS account to share your AWS resources
- ■ Create roles for your AWS account and define the users or services that can assume them
- ■ Use existing identities for your enterprise to grant permissions to perform tasks using AWS resources

By using IAM with EC2, you can control whether users in your organization can perform a task using specific EC2 API

actions and whether they can use specific AWS resources.

STORING AND MANAGING ENCRYPTION KEYS IN THE CLOUD



AWS Key Management Service ([AWS KMS](#)) is a managed service that makes it easy for you to create and control the encryption keys used to encrypt your data. The master keys that you create in AWS KMS are protected by FIPS 140-2 validated cryptographic modules.

AWS KMS is integrated with most other AWS services that encrypt your data with encryption keys that you manage. AWS KMS is also integrated with CloudTrail to provide encryption key usage logs to help meet your auditing, regulatory, and compliance needs.

You can perform the following management actions on your AWS KMS master keys:

- █ Create, describe, and list master keys
- █ Enable and disable master keys
- █ Create and view grants and access control policies for your master keys
- █ Enable and disable automatic rotation of the cryptographic material in a master key
- █ Import cryptographic material into an AWS KMS master key
- █ Tag your master keys for easier identification, categorizing, and tracking
- █ Create, delete, list, and update aliases, which are friendly names associated with your master keys
- █ Delete master keys to complete the key lifecycle

With AWS KMS, you can also perform the following

cryptographic functions using master keys:

- Encrypt, decrypt, and re-encrypt data
- Generate data encryption keys that you can export from the service in plaintext or encrypted under a master key that does not leave the service
- Generate random numbers suitable for cryptographic applications

By using AWS KMS, you gain more control over access to data you encrypt. You can use the key management and cryptographic features directly in your applications or through AWS services that are integrated with AWS KMS. Whether you are writing applications for AWS or using AWS services, AWS KMS enables you to maintain control over who can use your master keys and gain access to your encrypted data.

AWS KMS is integrated with CloudTrail, a service that delivers log files to an S3 bucket that you designate. By using CloudTrail, you can monitor and investigate how and when your master keys have been used and by whom.

PROTECTING DATA AT REST

Encryption is a powerful method to ensure that you protect data at rest within the AWS system. Because encryption keys are such an important component of this process, you need to consider the use of sophisticated tools like a key management infrastructure (KMI).

With AWS, there are three different models for how you and AWS provide the encryption of data at rest and the KMI. These three models are:

- You control the encryption method and the entire KMI.
- You control the encryption method, AWS provides the storage for KMI, and you control the management of the KMI.

- ■ AWS controls the encryption method and the entire KMI.

If you decide to control the encryption method and the entire KMI, you will use your own KMI to generate, store, and manage access to encryption keys. You will also, of course, control the encryption method used with your data at rest. You should note that the location of your KMI information and related security settings can be located on-premises or in AWS. Your encryption method can utilize proprietary, open-source, or a combination of tools. In fact, you might choose this model due to the full control you have over the encryption of your data at rest. AWS cannot encrypt or decrypt on your behalf. Once again, you have complete control.

The encryption you engage in with your data at rest might involve S3. You encrypt the data and then upload it to S3 using the S3 API. You can also use the AWS S3 encryption client. You supply the key to this client, and let it encrypt and decrypt the S3 data as part of your call to the S3 bucket.

You can opt to encrypt EBS volumes that instances use in EC2. Your approach might be at the block level or the file level.

You can encrypt data that you have on-premises and have this data stored in S3 through an AWS Storage Gateway.

You can encrypt your data stored in AWS RDS using a variety of approaches. You can also use your encryption in conjunction with AWS Elastic MapReduce. This is the Hadoop implementation in AWS.

Note

Many other services in AWS today offer encryption in addition to those mentioned here.

When you rely on AWS control for the encryption method and the entire KMI, you leverage AWS KMS. This is one option of many that are available to you.

DECOMMISSIONING DATA SECURELY

When a storage device has reached the end of its useful life, AWS procedures include a decommissioning process that is designed to prevent your data from being exposed to unauthorized individuals. AWS uses the techniques detailed in [NIST 800-88, “Guidelines for Media Sanitization,”](#) as part of the decommissioning process.

Per these guidelines, AWS personnel might take any sanitization actions necessary, including

- **Clear:** Applies logical techniques to sanitize data in all user-addressable storage locations for protection against simple noninvasive data recovery techniques; typically applied through the standard Read and Write commands to the storage device, such as by rewriting with a new value or using a menu option to reset the device to the factory state (where rewriting is not supported).
- **Purge:** Applies physical or logical techniques that render Target Data recovery infeasible using state-of-the-art laboratory techniques.
- **Destroy:** Renders Target Data recovery infeasible using state-of-the-art laboratory techniques and results in the subsequent inability to use the media for storage of data.

PROTECTING DATA IN TRANSIT

You can connect to an AWS access point via HTTP or [HTTPS](#) using Secure Sockets Layer (SSL). As you most likely know already, HTTPS is a cryptographic protocol that is designed to protect against eavesdropping, tampering, and message

forgery.

For customers who require additional layers of network security, AWS offers the Virtual Private Cloud (VPC), which provides a private subnet within the AWS cloud, and the ability to use an IPsec virtual private network (VPN) device to provide an encrypted tunnel between the VPC and your data center.

Remember also that many AWS customers opt for a Direct Connect private circuit for AWS connectivity of data in transit. This solution focuses on connectivity and not security. In fact, if encryption is required in this case, an IPsec tunnel still needs to be created across the Direct Connect circuit.

EXAM PREPARATION TASKS

As mentioned in the section “How to Use This Book” in the Introduction, you have a couple of choices for exam preparation: the exercises here, Chapter 16, “Final Preparation,” and the exam simulation questions in the Pearson Test Prep practice test software.

REVIEW ALL KEY TOPICS



Review the most important topics in this chapter, noted with the Key Topics icon in the margin of the page. Table 11-2 lists a reference of these key topics and the page numbers on which each is found.

Table 11-2 Key Topics for Chapter 11

Key Topic Element	Description	Page Number
List	IAM and EC2	181
Overview	AWS KMS	182

COMPLETE TABLES AND LISTS FROM MEMORY

Print a copy of Appendix C, “Memory Tables” (found on the book website), or at least the section for this chapter, and complete the tables and lists from memory. Appendix D, “Memory Tables Answer Key,” also on the website, includes completed tables and lists to check your work.

DEFINE KEY TERMS

Define the following key terms from this chapter and check your answers in the glossary:

AWS KMS

HTTPS

NIST

Q&A

The answer to this question appears in Appendix A. For more practice with exam format questions, use the Pearson Test

Prep practice test software.

- 1.** What are the three models for securing data at rest with encryption?

Chapter 12. Networking Infrastructure for a Single VPC Application

This chapter covers the following subjects:

- ■ **Introducing the Basic AWS Network Infrastructure:** This section introduces the components of networking in AWS from a broad overview perspective. It also introduces the default networking components that AWS provides in a Region.
- ■ **Network Interfaces:** You need to reach your resources in AWS. Network interfaces provide virtual constructs that emulate physical network cards; this permits connectivity. This section covers network interfaces in AWS.
- ■ **Route Tables:** What good is a network in AWS if there are no instructions for how to route packets from one resource to another? This is the job of your route tables. This section provides the basic design considerations for these important network components.
- ■ **Internet Gateways:** It is a fact of life that modern networks need Internet connectivity at some point. AWS Internet gateways make this possible. This section covers these gateways.
- ■ **Egress-Only Internet Gateways:** This section describes egress-only Internet gateways that might be required in your IPv6 AWS networks.
- ■ **DHCP Option Sets:** This section describes DHCP option sets and why you need them. It also provides guidance on those specific options supported in AWS.
- ■ **DNS:** AWS provides you with a DNS server for name resolution in your VPC. You can also use your own. This section covers these points.
- ■ **Elastic IP Addresses:** These special IP addresses allow you to easily make new resources available at an old address. This section provides important guidelines about these addresses.
- ■ **VPC Endpoints:** A VPC endpoint enables you to privately connect your VPC to supported AWS services and VPC endpoint services powered by

PrivateLink without requiring an Internet gateway, NAT device, VPN connection, or AWS Direct Connect connection. This section describes the two approaches you can use for this in AWS.

- ■ **NAT:** There are also two approaches to NAT in AWS. This section describes these approaches and provides AWS recommendations.
- ■ **VPC Peering:** To control communications between VPCs, you can use VPC peerings, which are described in this section.
- ■ **ClassicLink:** ClassicLink enables you to link your EC2-Classic instance to a VPC in your account within the same region. This section describes ClassicLink in more detail.

Network components are just as important in the cloud infrastructure as they are in your traditional infrastructure. This chapter examines the network infrastructure components that you might incorporate into an AWS application.

“DO I KNOW THIS ALREADY?” QUIZ

The “Do I Know This Already?” quiz allows you to assess whether you should read the entire chapter. Table 12-1 lists the major headings in this chapter and the “Do I Know This Already?” quiz questions covering the material in those headings so you can assess your knowledge of these specific areas. The answers to the “Do I Know This Already?” quiz appear in Appendix A.

Table 12-1 “Do I Know This Already?” Foundation Topics
Section-to-Question Mapping

Foundation Topics Section	Questions
Network Interfaces	1

Route Tables	2
Internet Gateways	3
Egress-Only Internet Gateways	4
DHCP Option Sets	5
DNS	6
Elastic IP Addresses	7
VPC Endpoints	8
NAT	9
VPC Peering	10

Caution

The goal of self-assessment is to gauge your mastery of the topics in this chapter. If you do not know the answer to a question or are only partially sure of the answer, you should mark that question as wrong for purposes of the self-assessment. Giving yourself credit for an answer you correctly guess skews your self-assessment results and might provide you with a false sense of security.

- 1.** What is true regarding network interfaces in AWS?
 - a.** When you detach a network interface and attach it to a new instance, the attributes are reset.
 - b.** There is no default network interface for an instance.
 - c.** You cannot detach a primary network interface.
 - d.** You cannot assign more than one network interface to an instance.
- 2.** What happens to subnets that are not assigned a route table?
 - a.** They are terminated.
 - b.** They are associated with the main route table.
 - c.** They are deactivated.
 - d.** Their creation fails with an error.
- 3.** What component is the target of the default route in the main route table for your default VPC?
 - a.** Internet Gateway
 - b.** DNS
 - c.** Egress-Only Internet Gateway

d.NAT Gateway

4. For what protocol is the Egress-Only Internet Gateway designed?

a.STP

b.OSPF

c.BGP

d.IPv6

5. Which is not a supported DHCP Option with AWS?

a. file-servers

b. ntp-servers

c. domain-name

d. domain-name-servers

6. How would you configure AWS to use your own DNS server?

a.By configuring a local forwarder

b.By using a DHCP option set

c.By deleting the default DNS server

d.By configuring a NAT Gateway

7. Which of the following is not true regarding Elastic IP addresses?

a.You can keep a disassociated Elastic IP address in your account.

b.AWS resolves a public DNS name to the IPv4 address.

c.Elastic IP addresses can be used across Regions.

d.You must allocate one to your account first before you

can use it.

8. Which are types of VPC endpoints? (Choose two.)

- a. Public endpoints
- b. Interface endpoints
- c. Private endpoints
- d. Gateway endpoints

9. What is the recommended NAT implementation in AWS?

- a. NAT Gateway
- b. NAT Instance
- c. NAT Proxy
- d. NAT Controller

10. Which is not a valid use of a VPC peering?

- a. To connect two VPCs in the same account
- b. To connect two VPCs in two different Regions
- c. To connect two VPCs in two different accounts
- d. To connect two AZs within a VPC

11. What is ClassicLink used for?

- a. S3
- b. DNS
- c. NAT
- d. EC2-Classic

FOUNDATION TOPICS

INTRODUCING THE BASIC AWS NETWORK INFRASTRUCTURE

The Virtual Private Cloud is the main networking architecture of your AWS implementation. As you might guess, the VPC is made up of many important components.

This chapter covers these components in detail, but here is a quick summary of each that we examine:



- **■ Network Interfaces:** This logical network component serves to represent a physical network interface card (NIC); as such, this component can be configured with IPv4 and IPv6 addresses.
- **■ Route Tables:** Just as would exist on a physical router, AWS route tables contain a set of rules, called routes, that are used to determine where network traffic is directed.
- **■ Internet Gateways:** An Internet gateway serves two purposes: to provide a target in your VPC route tables for Internet-routable traffic and to perform Network Address Translation (NAT) for instances that have been assigned public IPv4 addresses.
- **■ Egress-Only Internet Gateways:** These VPC components allow outbound communication over IPv6 from instances in your VPC to the Internet and prevent the Internet from initiating an IPv6 connection with your instances.
- **■ DHCP Option Sets:** DHCP provides a standard for passing configuration information to hosts on a TCP/IP network. The Options field of a DHCP message contains the configuration parameters. Some of those parameters are the domain name, domain name server, and the netbios-node-type. The option sets allow you to configure such options for your VPC.
- **■ DNS:** AWS provides you with a DNS server for your VPC, but it is important to realize that you can also use your own.
- **■ Elastic IP Addresses:** These static IPv4 addresses are designed for dynamic cloud computing. An Elastic IP address is associated with your AWS account; with this address, you can mask the failure of an instance or

software by rapidly remapping the address to another instance in your account.

- **VPC Endpoints:** These endpoints enable you to privately connect your VPC to supported AWS services and VPC endpoint services powered by PrivateLink without requiring an Internet gateway, NAT device, VPN connection, or AWS Direct Connect connection.
- **NAT:** You can use a NAT device to enable instances in a private subnet to connect to the Internet (for example, for software updates) or other AWS services but prevent the Internet from initiating connections with the instances. AWS offers two kinds of NAT devices—a NAT gateway and a NAT instance—but strongly recommends the use of NAT gateways.
- **VPC Peering:** This networking connection between two VPCs enables you to route traffic between them privately. You can create a VPC peering connection between your own VPCs, with a VPC in another AWS account, or with a VPC in a different AWS Region.
- **ClassicLink:** This component allows you to link your EC2-Classic instance to a VPC in your account, within the same region. This allows you to associate the VPC security groups with the EC2-Classic instance, enabling communication between your EC2-Classic instance and instances in your VPC using private IPv4 addresses.

Lab: Checking Default Networking Components in a Region

In the following steps, you examine the networking components configured for you by default in AWS:

Step 1. Log in to the AWS Management Console. From the Region drop-down menu, select a Region you have made no configurations in. For this example, I used the Asia Pacific (Mumbai) Region.

Step 2. From the list of services, select **VPC**. Figure 12-1 shows some of the default components created for you for networking in AWS. The complete list is as follows:

- One default VPC
- Two default subnets
- One default route table
- One default Internet gateway
- One default DHCP option set
- One default network ACL
- One default security group

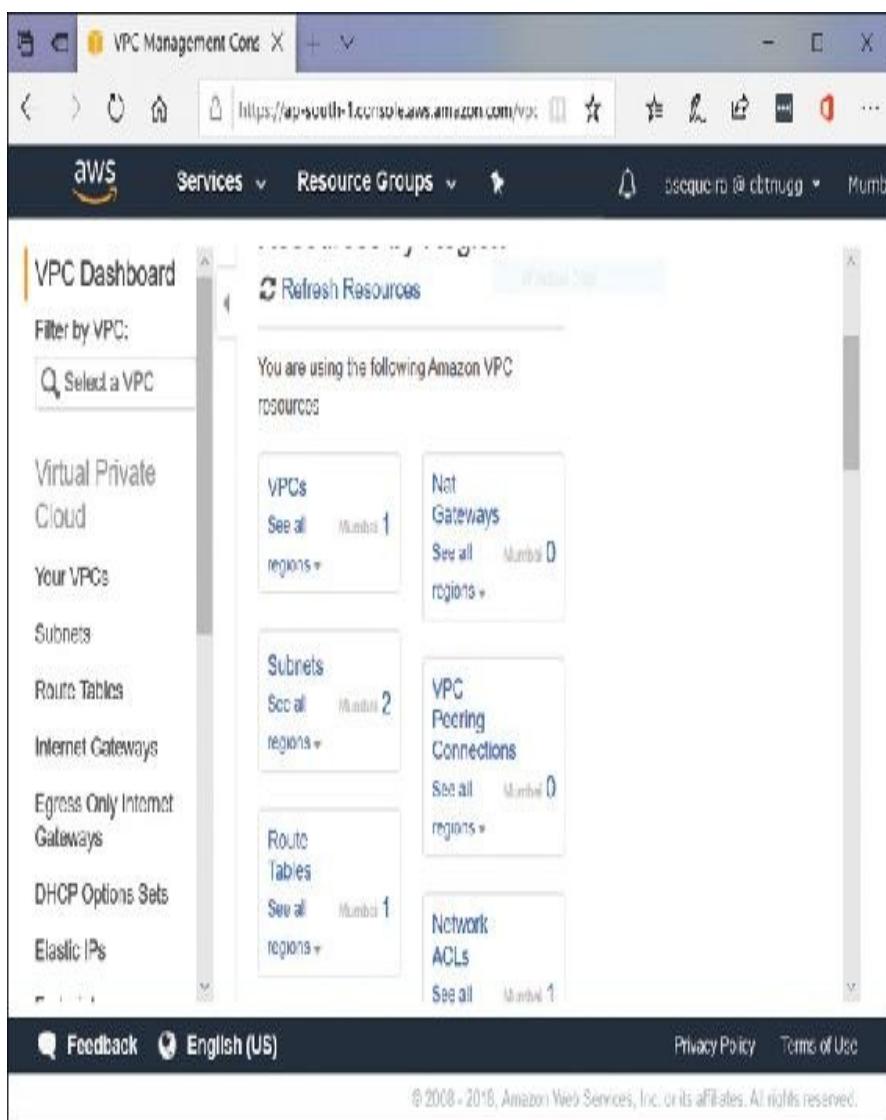


Figure 12-1 The Default VPC Components in a Region

Step 3. Select the **1** hyperlink to explore the default VPC created for your region. Note the following parameters:

- ■ The VPC is given a unique VPC ID.
- ■ The default state is Available.
- ■ The VPC is assigned an IPv4 CIDR block (private IP address space).
- ■ The VPC is associated with the default DHCP option set.
- ■ The VPC is associated with the default route table.
- ■ The VPC is associated with the default network ACL.
- ■ The Tenancy mode is the default.
- ■ The VPC is flagged as the default VPC.

Step 4. To examine the default subnets, click the **Subnet** link in the left column of the Management Console.

Figure 12-2 shows the results for the default subnets in the default VPC. Note the following:

- ■ Each subnet is assigned a unique Subnet ID.
- ■ Each subnet defaults to Available.
- ■ Each subnet is associated with the default VPC.
- ■ Each subnet has its own portion of the privately addressed CIDR block.
- ■ Each subnet has 4091 available IP addresses by default.
- ■ Each subnet is placed in a different Availability Zone.
- ■ Each subnet is associated with the default route table.
- ■ Each subnet is associated with the default network ACL.
- ■ Each subnet is flagged as a default subnet.
- ■ Each subnet is set to auto assign public IP addressing.

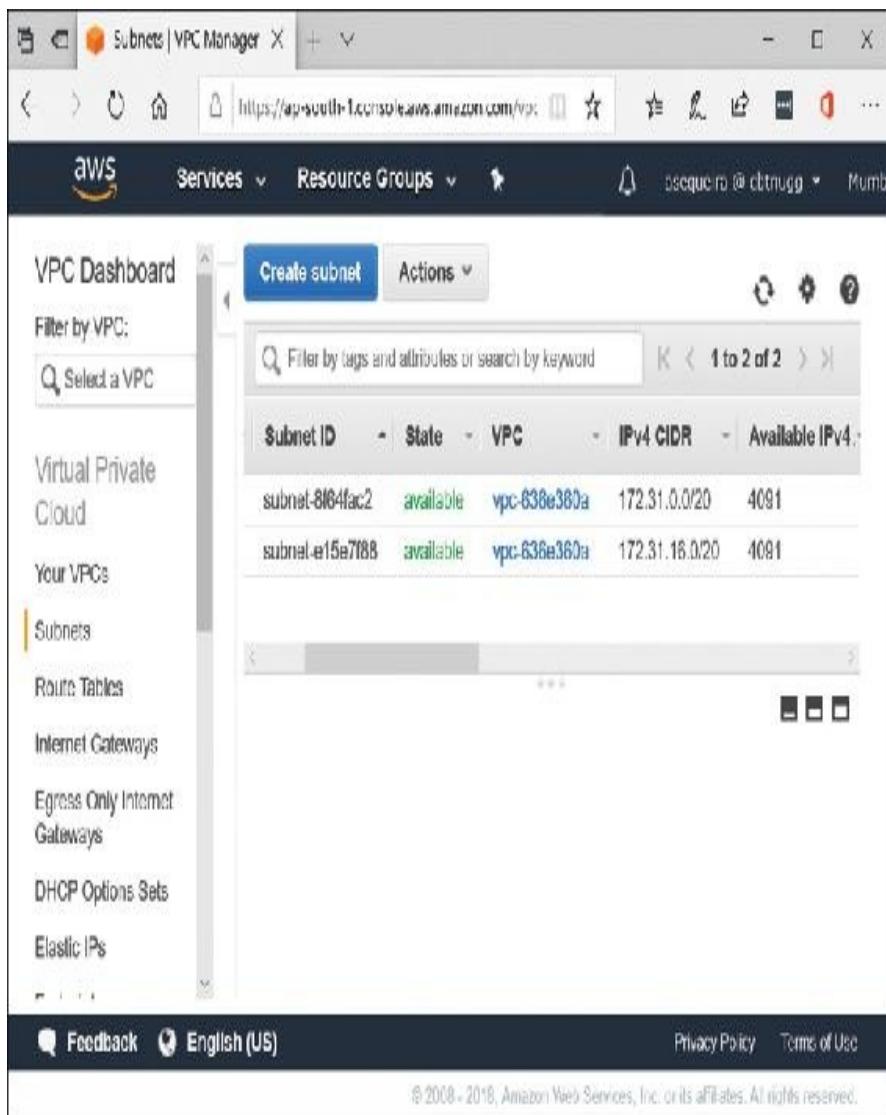


Figure 12-2 The Default Subnet Properties

Step 5. For one of the default subnets, click the hyperlink for the **route table ID** to view the default route table. Click the check box next to this default route table to view its additional properties. Note the following:

- The default route table is provided a unique route table ID.
- Zero subnets are explicitly associated with this route table; your default subnets are implicitly associated with the route table.

- ■ The route table is flagged as the main route table for the VPC listed in the VPC column.
- ■ Under the Routes tab, there is a route for local IPv4 addresses in the private CIDR block, and there is also a default route (0.0.0.0/0) pointing to the default Internet gateway for all other prefix destinations.
- ■ Under the Subnet Associations tab, the default subnets are indeed associated with the default route table.
- ■ Under the Route Propagation tab, no virtual private gateways are permitted to update this table.
- ■ No tags are associated with the default route table.

Step 6. Select the **Routes** tab in the Subnets area and click the target hyperlink that is the unique ID of your Internet gateway to view its properties. Note the following:

- ■ The default Internet gateway is assigned a unique ID.
- ■ The default state is Attached.
- ■ The VPC association is with the default VPC.

Step 7. In the column on the left, select **DHCP Option Sets** to examine your defaults. Choose the check box next to your default DHCP option set to view additional properties. Note the following:

- ■ The default DHCP option set is given a unique ID.
- ■ The default options actually set are as follows:

```
domain-name = ap-south-1.compute.internal
domain-name-servers = AmazonProvidedDNS
```

Step 8. In the column on the left, select **Network ACLs** to examine the default network ACL. Click the check box next to this component to view additional

properties. Note the following:

- ■ The network ACL is given a unique ID.
- ■ The network ACL is associated with the two default subnets by default.
- ■ The network ACL is flagged as the default.
- ■ Under the Inbound Rules tab, all traffic is permitted.
- ■ Under the Outbound Rules tab, all traffic is permitted.

Step 9. In the column on the left, select **Security Groups**

to examine the default security group in place in your VPC. Click the check box next to the default security group to examine its additional properties.

Note the following:

- ■ The security group is assigned a unique ID.
- ■ The Group Name is default.
- ■ Under the Inbound Rules tab, all traffic is permitted.
- ■ Under the Outbound Rules tab, all traffic is permitted.

NETWORK INTERFACES

An AWS network interface is a virtual network interface that can include the following attributes:

- ■ A primary private IPv4 address
- ■ One or more secondary private IPv4 addresses
- ■ One Elastic IP address per private IPv4 address
- ■ One public IPv4 address, which can be auto-assigned to the network interface for eth0 when you launch an instance
- ■ One or more IPv6 addresses
- ■ One or more security groups
- ■ A MAC address
- ■ A source/destination check flag

- ■ A description

Keep the following in mind regarding network interfaces in AWS:

Key Topic

- ■ You can create a network interface, attach it to an instance, detach it from an instance, and attach it to another instance.
- ■ A network interface's attributes follow it as it is attached or detached from an instance and reattached to another instance.
- ■ When you move a network interface from one instance to another, network traffic is redirected to the new instance.
- ■ Each instance in your VPC has a default network interface (the primary network interface) that is assigned a private IPv4 address from the IPv4 address range of your VPC.
- ■ You cannot detach a primary network interface from an instance. You can create and attach an additional network interface to any instance in your subnet.
- ■ The number of network interfaces you can attach varies by instance type. Attaching multiple network interfaces to an instance is useful when you want to create a management network, use network and security appliances in your VPC, create dual-homed instances with workloads/roles on distinct subnets, or create a low-budget, high-availability solution.

ROUTE TABLES

A route table contains a set of rules that are used to determine where network traffic is directed. Each subnet in your VPC must be associated with a route table; the table controls the routing for the subnet. A subnet can be associated with only one route table at a time, but you can associate multiple subnets with the same route table.

When you create a VPC, it automatically has a main route table. The main route table controls the routing for all subnets

that are not explicitly associated with any other route table. You can add, remove, and modify routes in the main route table.

You can explicitly associate a subnet with the main route table even if it is already implicitly associated. You might do that if you change which table is the main route table, which changes the default for additional new subnets, or any subnets that are not explicitly associated with any other route table.



Keep these points in mind regarding route tables in AWS:

- ■ Your VPC has an implicit router.
- ■ Your VPC automatically comes with a main route table that you can modify.
- ■ You can create additional custom route tables for your VPC.
- ■ Each subnet must be associated with a route table, which controls the routing for the subnet. If you do not explicitly associate a subnet with a particular route table, the subnet is implicitly associated with the main route table.
- ■ You cannot delete the main route table, but you can replace the main route table with a custom table that you have created.
- ■ Each route in a table specifies a destination CIDR and a target. Just like physical routers, the most specific route that matches the traffic determines how to route the traffic.
- ■ CIDR blocks for IPv4 and IPv6 are treated separately.
- ■ Every route table contains a local route for communication within the VPC over IPv4. If your VPC has more than one IPv4 CIDR block, your route tables contain a local route for each IPv4 CIDR block. If you have associated an IPv6 CIDR block with your VPC, your route tables contain a local route for the IPv6 CIDR block. You cannot modify or delete these routes.
- ■ When you add an Internet gateway, an egress-only Internet gateway, a

virtual private gateway, a NAT device, a peering connection, or a VPC endpoint in your VPC, you must update the route table for any subnet that uses these gateways or connections.

- There is a limit on the number of route tables you can create per VPC and the number of routes you can add per route table.

Figure 12-3 shows the makeup of the default route table in a default VPC.

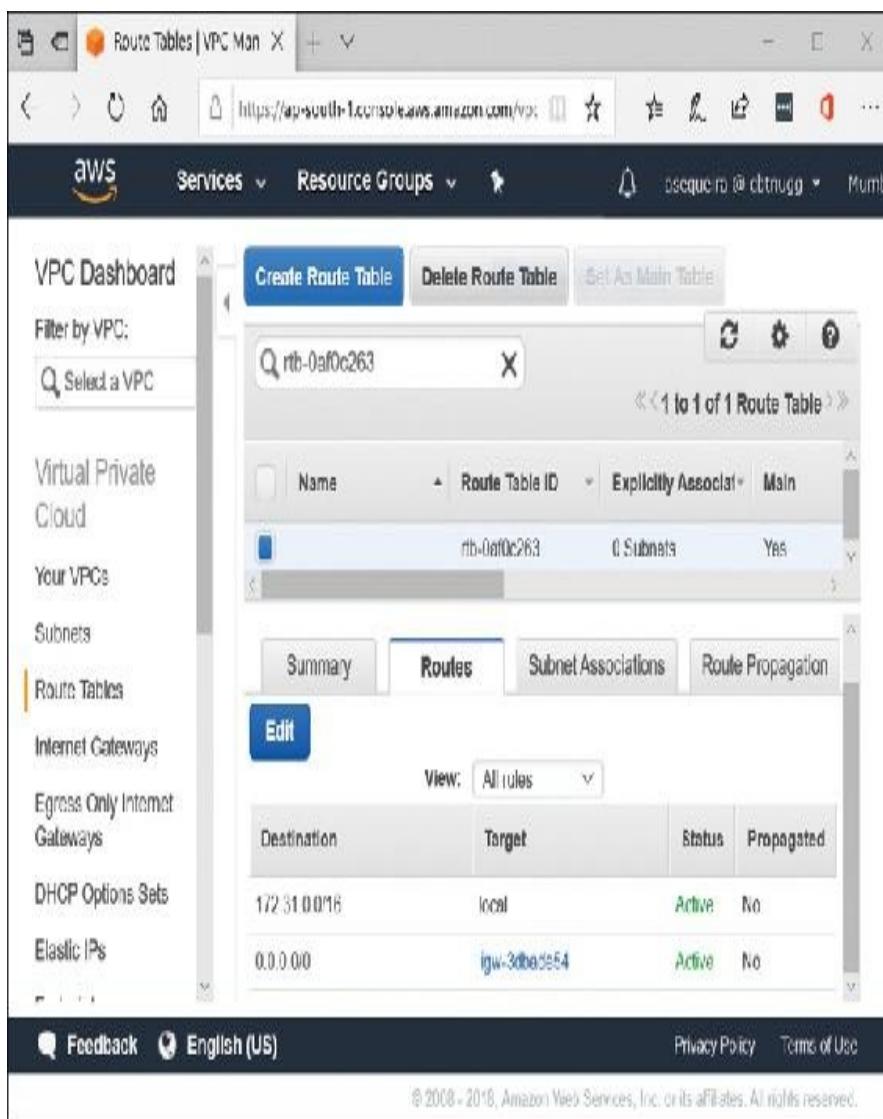


Figure 12-3 The Default Route Table in the Default VPC

INTERNET GATEWAYS

An Internet gateway is a highly available VPC component that allows communication between instances in your VPC and the Internet. It therefore imposes no availability risks or bandwidth constraints on your network traffic.

An Internet gateway serves two purposes: to provide a target in your VPC route tables for Internet-routable traffic and to perform Network Address Translation (NAT) for instances that have been assigned public IPv4 addresses. An Internet gateway supports IPv4 and IPv6 traffic.



To enable access to or from the Internet for instances in a VPC subnet, you must do the following:

- Attach an Internet gateway to your VPC.
- Ensure that your subnet's route table points to the Internet gateway.
- Ensure that instances in your subnet have a globally unique IP address.
- Ensure that your network access control and security group rules allow the relevant traffic to flow to and from your instance.

You can scope the route to all destinations not explicitly known to the route table (0.0.0.0/0 for IPv4 or ::/0 for IPv6), or you can scope the route to a narrower range of IP addresses. If your subnet is associated with a route table that has a route to an Internet gateway, it is known as a public subnet.

EGRESS-ONLY INTERNET GATEWAYS

An egress-only Internet gateway is a highly available VPC component that allows outbound communication over IPv6 from instances in your VPC to the Internet and prevents the Internet from initiating an IPv6 connection with your

instances.

Note

To enable outbound-only Internet communication over IPv4, use a NAT gateway instead.

IPv6 addresses are globally unique and are therefore public by default. If you want your instance to be able to access the Internet but you want to prevent resources on the Internet from initiating communication with your instance, you can use this egress-only Internet gateway.

To do this, create an egress-only Internet gateway in your VPC and then add a route to your route table that points all IPv6 traffic (::/0) or a specific range of IPv6 addresses to the egress-only Internet gateway. IPv6 traffic in the subnet that is associated with the route table is routed to the egress-only Internet gateway.

An egress-only Internet gateway is stateful: it forwards traffic from the instances in the subnet to the Internet or other AWS services and then sends the response back to the instances.

An egress-only Internet gateway has the following characteristics:

- ■ You cannot associate a security group with an egress-only Internet gateway. You can use security groups for your instances in the private subnet to control the traffic to and from those instances.
- ■ You can use a network ACL to control the traffic to and from the subnet for which the egress-only Internet gateway routes traffic.

DHCP OPTION SETS

The EC2 instances you launch into a nondefault VPC are private by default; they are not assigned a public IPv4 address unless you specifically assign one during launch, or you modify the subnet's public IPv4 address attribute. By default,

all instances in a nondefault VPC receive a hostname that AWS assigns (for example, ip-10-0-0-202). You can assign your own domain name to your instances and use up to four of your own DNS servers. To do that, you must specify a special set of DHCP options to use with the VPC. DHCP option sets in AWS permit this configuration.

Supported options include



- **domain-name-servers**
- **domain-name**
- **ntp-servers**
- **netbios-name-servers**
- **netbios-node-type**

DNS

As you know, Domain Name System (DNS) is a standard by which names used on the Internet are resolved to their corresponding IP addresses. A DNS hostname is a name that uniquely and absolutely names a computer. It is composed of a hostname and a domain name. DNS servers resolve DNS hostnames to their corresponding IP addresses.

Public IPv4 addresses enable communication over the Internet, while private IPv4 addresses enable communication within the network of the instance (either EC2-Classic or a VPC).

AWS provides you with an Amazon DNS server. To use your own DNS server, create a new set of DHCP options for your VPC.

ELASTIC IP ADDRESSES

An Elastic IP address is a static IPv4 address designed for dynamic cloud computing. With an Elastic IP address, you can mask the failure of an instance or software by rapidly remapping the address to another instance in your account.

An Elastic IP address is a public IPv4 address, which is reachable from the Internet. If your instance does not have a public IPv4 address, you can associate an Elastic IP address with your instance to enable communication with the Internet; for example, to connect to your instance from your local computer. AWS does not currently support Elastic IP addresses for IPv6.



The following are the basic characteristics of an Elastic IP address:

- To use an Elastic IP address, you first allocate one to your account and then associate it with your instance or a network interface.
- When you associate an Elastic IP address with an instance or its primary network interface, the instance's public IPv4 address (if it had one) is released back into Amazon's pool of public IPv4 addresses. You cannot reuse a public IPv4 address.
- You can disassociate an Elastic IP address from a resource and reassociate it with a different resource. Any open connections to an instance continue to work for a time even after you disassociate its Elastic IP address and reassociate it with another instance. We recommend that you reopen these connections using the reassigned Elastic IP address.
- A disassociated Elastic IP address remains allocated to your account until you explicitly release it.
- To ensure efficient use of Elastic IP addresses, Amazon imposes a small hourly charge if an Elastic IP address is not associated with a running instance, or if it is associated with a stopped instance or an unattached

network interface. While your instance is running, you are not charged for one Elastic IP address associated with the instance, but you are charged for any additional Elastic IP addresses associated with the instance.

- ■ An Elastic IP address is for use in a specific region only.
- ■ When you associate an Elastic IP address with an instance that previously had a public IPv4 address, the public DNS hostname of the instance changes to match the Elastic IP address.
- ■ AWS resolves a public DNS hostname to the public IPv4 address or the Elastic IP address of the instance outside the network of the instance, and to the private IPv4 address of the instance from within the network of the instance.

VPC ENDPOINTS

A VPC endpoint enables you to privately connect your VPC to supported AWS services and VPC endpoint services powered by PrivateLink without requiring an Internet gateway, NAT device, VPN connection, or AWS Direct Connect connection. Instances in your VPC do not require public IP addresses to communicate with resources in the service. Traffic between your VPC and the other service does not leave the Amazon network.

Endpoints are virtual devices. They are highly available VPC components that allow communication between instances in your VPC and services without imposing availability risks or bandwidth constraints on your network traffic.

There are two types of VPC endpoints: interface endpoints and gateway endpoints. You should create the type of VPC endpoint required by the supported service.



Interface Endpoints (Powered by AWS PrivateLink)

An interface endpoint is an elastic network interface with a private IP address that serves as an entry point for traffic destined to a supported service. The following services are supported:

-  API Gateway
-  CloudWatch
-  CloudWatch Events
-  CloudWatch Logs
-  CodeBuild
-  Config
-  EC2 API
-  Elastic Load Balancing API
-  Key Management Service
-  Kinesis Data Streams
-  SageMaker Runtime
-  Secrets Manager
-  Security Token Service
-  Service Catalog
-  SNS
-  Systems Manager
-  Endpoint services hosted by other AWS accounts
-  Supported AWS Marketplace partner services

Gateway Endpoints

A gateway endpoint is a gateway that is a target for a specified route in your route table, used for traffic destined to a supported AWS service. The following AWS services are supported:

- S3
- DynamoDB

NAT

You can use a NAT device to enable instances in a private subnet to connect to the Internet or other AWS services but prevent the Internet from initiating connections with the instances. A NAT device forwards traffic from the instances in the private subnet to the Internet or other AWS services and then sends the response back to the instances.

When traffic goes to the Internet, the source IPv4 address is replaced with the NAT device's address, and similarly, when the response traffic goes to those instances, the NAT device translates the address back to those instances' private IPv4 addresses.

NAT devices are not supported for IPv6 traffic. You should use an egress-only Internet gateway instead.

AWS offers two kinds of NAT devices: a NAT gateway and a NAT instance. AWS recommends NAT gateways because they provide better availability and bandwidth over NAT instances. The NAT Gateway service is also a managed service that does not require your administration efforts.

A NAT instance is launched from a NAT AMI. You might choose to use a NAT instance for special purposes.

VPC PEERING

An AWS VPC peering connection is a networking connection between two VPCs that enables you to route traffic between them privately. Instances in either VPC can communicate with

each other as if they are within the same network. You can create a VPC peering connection between your own VPCs, with a VPC in another AWS account, or with a VPC in a different AWS Region.

AWS uses the existing infrastructure of a VPC to create a VPC peering connection; it is neither a gateway nor a VPN connection, and does not rely on a separate piece of physical hardware. There is no single point of failure for communication or a bandwidth bottleneck.

CLASSICLINK

ClassicLink allows you to link your EC2-Classic instance to a VPC in your account, within the same region. This enables you to associate the VPC security groups with the EC2-Classic instance, enabling communication between your EC2-Classic instance and instances in your VPC using private IPv4 addresses.

ClassicLink removes the need to make use of public IPv4 addresses or Elastic IP addresses to enable communication between instances in these platforms. It is available to all users with accounts that support the EC2-Classic platform and can be used with any EC2-Classic instance.

There is no additional charge for using ClassicLink. Standard charges for data transfer and instance usage apply.

Note

EC2-Classic instances cannot be enabled for IPv6 communication. You can associate an IPv6 CIDR block with your VPC and assign IPv6 addresses to resources in your VPC; however, communication between a ClassicLinked instance and resources in the VPC is over IPv4 only.

EXAM PREPARATION TASKS

As mentioned in the section “How to Use This Book” in the Introduction, you have a couple of choices for exam preparation: the exercises here, Chapter 16, “Final Preparation,” and the exam simulation questions in the Pearson Test Prep practice test software.

REVIEW ALL KEY TOPICS

Review the most important topics in this chapter, noted with the Key Topics icon in the margin of the page. Table 12-2 lists a reference of these key topics and the page numbers on which each is found.

Table 12-2 Key Topics for Chapter 12

Key Topic Element	Description	Page Number
List	VPC network components	194
List	Network interface guidelines	199
List	Route table guidelines	200
List	Support Internet access for instances	202

List	Supported DHCP options	203
List	Characteristics of Elastic IP addresses	204
Concept	Two types of VPC endpoints	205

COMPLETE TABLES AND LISTS FROM MEMORY

There are no memory tables in this chapter.

DEFINE KEY TERMS

Define the following key terms from this chapter and check your answers in the glossary:

Network interface

Route table

Internet gateway

Egress-only Internet gateway

DHCP option sets

DNS

Elastic IP addresses

VPC endpoints

NAT

VPC peering

Q&A

The answers to these questions appear in [Appendix A](#). For more practice with exam format questions, use the Pearson Test Prep practice test software.

- 1.** What VPC component allows you to configure communications between two different VPCs residing in two different accounts?
- 2.** What VPC component allows you to assign unique DNS server addresses to your instances?
- 3.** What VPC component allows you to configure IPv6 access from instances to the Internet but does not permit outside connections to be established to your VPC?
- 4.** What VPC component is associated with all of your subnets and provides instructions on how to direct packets?

Part IV

Domain 4: Design Cost-Optimized Architectures

Chapter 13. Cost-Optimized Storage

This chapter covers the following subjects:

- **S3 Services:** There are a variety of ways to save money in your S3 usage. This section describes how you are charged for S3 and provides ideas on cost savings.
- **EBS Services:** This section ensures you understand the charges associated with EBS.
- **EFS Services:** This section describes the potential for costs associated with EFS.
- **Glacier Services:** This section describes the methods that are used to calculate your Glacier charges.

AWS cloud storage presents amazing capabilities when it comes to durability and scalability. But what about costs? This chapter focuses on the charges you might incur when it comes to your various cloud-based storage solutions.

“DO I KNOW THIS ALREADY?” QUIZ

The “Do I Know This Already?” quiz allows you to assess whether you should read the entire chapter. Table 13-1 lists the major headings in this chapter and the “Do I Know This Already?” quiz questions covering the material in those headings so you can assess your knowledge of these specific areas. The answers to the “Do I Know This Already?” quiz appear in Appendix A.

Table 13-1 “Do I Know This Already?” Foundation Topics
Section-to-Question Mapping

Foundation Topics Section	Questions
S3 Services	1–2
EBS Services	3–4
EFS Services	5–6
Glacier Services	7–8

Caution

The goal of self-assessment is to gauge your mastery of the topics in this chapter. If you do not know the answer to a question or are only partially sure of the answer, you should mark that question as wrong for purposes of the self-assessment. Giving yourself credit for an answer you correctly guess skews your self-assessment results and might provide you with a false sense of security.

- 1.** Which of the following is not a metric that you are charged for with S3?
 - a.** Number of unique object IDs consumed
 - b.** Data requests
 - c.** Network Data Transferred In
 - d.** Network Data Transferred Out

2. Which of the following would result in a transfer fee with S3?
- a. Data from EC2 in one Region to S3 in the same Region
 - b. Data from one bucket in a Region to another bucket in the same Region
 - c. Data from one Region to another
 - d. A transfer from folder to folder inside a bucket
3. Which of the following is not a form of EBS storage?
- a. Gen 3 SSD
 - b. Provisioned IOPS SSD
 - c. Throughput Optimized HDD
 - d. Cold HDD
4. You have disconnected an EBS volume that uses Provisioned IOPS SSD. Which statement is true?
- a. You are still incurring charges for this volume.
 - b. You will incur charges the moment the volume is attached to another EC2 instance.
 - c. You will incur charges when the volume is rebooted.
 - d. You will incur charges only if the volume is moved to another Region.
5. What is the recommended method of monitoring the amount of data that you are moving with EFS File Sync?
- a. ElastiCache
 - b. CloudTrail
 - c. CloudFormation

d. CloudWatch

- 6.** What is the baseline rate for EFS under which there are no additional charges?
- a.** 20 KB/s per gigabyte of throughput
 - b.** 246 KB/s per gigabyte of throughput
 - c.** 100 KB/s per gigabyte of throughput
 - d.** 50 KB/s per gigabyte of throughput
- 7.** What is the required minimum duration for a Glacier archive?
- a.** 30 days
 - b.** 90 days
 - c.** 120 days
 - d.** 60 days
- 8.** Which is not a retrieval tier of Glacier?
- a.** Expedited retrieval
 - b.** Standard retrieval
 - c.** Bulk retrieval
 - d.** Mission-critical retrieval

FOUNDATION TOPICS

S3 SERVICES

You already realize that one of the most incredible features of Amazon S3 is that you pay only for storage costs associated with your use of S3. There are no minimum fees. This lack of

up-front costs allows you to scale your storage as you can afford to do so.

Lab: Estimating AWS S3 Costs



There is a simple way to estimate your costs in AWS S3: use the [AWS Simple Monthly calculator](#). The steps that follow demonstrate some facts about S3 costs.

Step 1. Use Google or your favorite search engine to search for AWS Simple Monthly Calculator. Click on the link to access the online calculator. [Figure 13-1](#) show the AWS Simple Monthly Calculator main page.

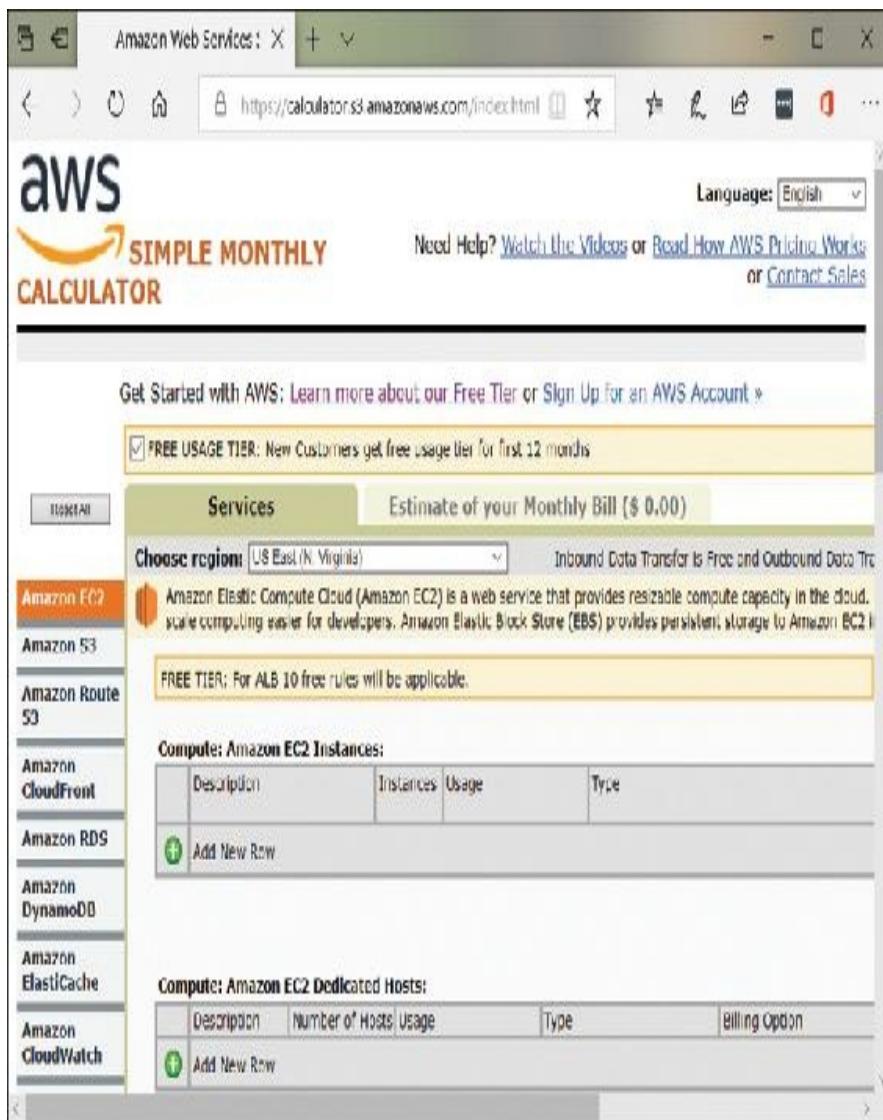


Figure 13-1 The AWS Simple Monthly Calculator

Step 2. On the left side, click the **Amazon S3** link.

Step 3. Amazon S3 costs can vary by Region. Select the **US West (Oregon)** Region from the drop-down menu. Also, above this area, make sure that Free Tier is unchecked. This will give you estimates that do not include any free tier levels of access.

Step 4. Under S3 Standard Storage & Requests, enter **800 GB** as an example. The amount of your estimated

monthly charge for this storage appears in a tab near the top of the interface. In this example (yours might vary because AWS charges also vary over time), this results in approximately \$18.40 per month in fees.

Step 5. Choose the **Clear Form** button to clear this entry.

Step 6. To examine how to save if you choose a different S3 storage tier, under the S3 One Zone-Infrequent Access (S3 One Zone-IA) Storage & Requests field, enter **800 GB** and examine the result. In this case, the result is just \$8 per month!

Note that a choice of Region could impact your S3 costs. Remember, your AWS S3 bucket name is a global resource, but when you are creating your bucket, you choose a Region for its storage. AWS charges you less where the actual storage costs are less.

Following are other facts to keep in mind regarding S3 storage costs:

- ■ There is no data transfer charge for data transferred within a Region. There is only a charge for such a transfer when it is between Regions.
- ■ There is no data transfer charge for data transferred between EC2 and S3 in the same Region.
- ■ There is no data transfer charge for data transferred between the EC2 Northern Virginia Region and S3 US East (Northern Virginia) Region. These are effectively the same Region; thus, there is no charge.
- ■ S3 is part of the free tier. You can get started for free in all Regions except the AWS GovCloud Region. You receive 5 GB of S3 Standard for free, including 20,000 GET Requests, 2000 PUT Requests, 15 GB of data transfer in, and 15 GB of data transfer out each month for one year.

It is important to know how you are charged for AWS S3 storage if you are outside the free tier. You should be aware of

the following charges:

- **Storage Used:** This charge is based on average storage throughout the month.
- **Network Data Transferred In:** This charge represents the amount of data sent to your S3 buckets. Although this information is tracked in AWS, there is no charge for it.
- **Network Data Transferred Out:** This charge applies whenever data is read from any of your buckets outside the given S3 Region.
- **Data Requests:** These charges include PUT requests, GET requests, and DELETE requests.
- **Data Retrieval:** These charges apply to the S3 Standard-Infrequent Access and S3 One Zone-IA storage classes.

What about versioning charges in S3? Keep in mind that S3 storage rates also apply to the versions of files you keep.

Although versioning is excellent for fault tolerance and backup strategies, it can dramatically increase storage costs.

You can control costs by automating the movement of objects in S3 to different storage tiers using Lifecycle Management.

While this is excellent news for your cost controls with AWS, even more good news is the fact that Lifecycle Management in AWS is free of charge!

Lab: Implementing Lifecycle Management

The following steps ensure you understand how easy it is configure Lifecycle Management using the AWS Management Console:

Step 1. Open the AWS Management Console and search for the S3 service. Click the link.

Step 2. Click **Create Bucket**.

Step 3. For Bucket Name, enter your last name followed

by three random numbers.

Step 4. For Region, choose **US West (Oregon)** and click **Next**.

Step 5. Click **Next** on the Properties window to accept all defaults.

Step 6. Choose **Next** to accept the default permissions.

Step 7. Click **Create Bucket** on the Review screen.

Step 8. Select your bucket from the list and click the **Management** tab.

Step 9. Click the **Add Lifecycle Rule** button.

Step 10. Name your rule **MyRule** and click **Next**.

Step 11. Choose **Current Version** to configure a transition.

Step 12. Choose **Add Translation**.

Step 13. Under Object Creation, choose **Transition to One Zone-IA after** and leave the default 30 days.

Figure 13-2 shows this screen.

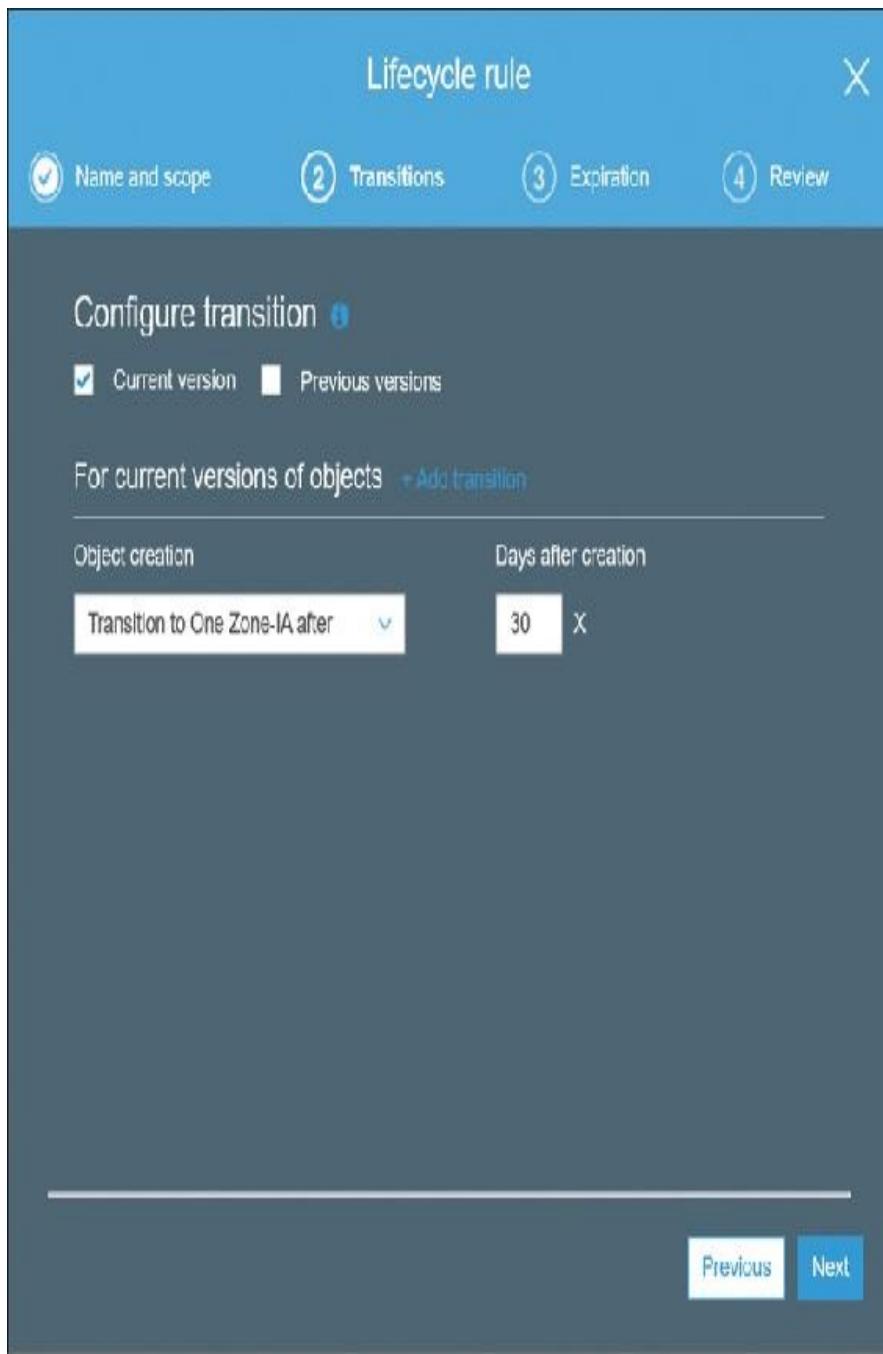


Figure 13-2 Configuring Lifecycle Management in AWS S3

Step 14. Choose **Next**.

Step 15. Choose **Next** at the Configure Expiration window.

Step 16. Choose **Save** at the Review screen.

Lab Cleanup

Step 1. Navigate to the S3 console in the AWS Management Console.

Step 2. Highlight the bucket you created in this lab.

Step 3. Click the **Delete Bucket** button.

Step 4. Type the name of the bucket and click **Confirm**.

EBS SERVICES

Just like with S3, you have the luxury of paying for only what you need when it comes to EBS storage. Also, like S3, prices for EBS vary based on the Region you're in.

Unlike S3, however, with EBS services, you pay for provisioned storage, not just used storage. Overprovisioning storage can add up across an infrastructure. Remember that it is better to grow the EBS volume as time goes on and minimize white space (unless the white space is needed for IOPS requirements).

Table 13-2 provides examples of pricing in the US West (Oregon) Region.

Table 13-2 EBS Pricing Examples for US West (Oregon)

Volume Type	Price
General-Purpose SSD	\$0.10 per GB per month

Provisioned IOPS SSD	\$0.125 per GB per month and \$0.065 per provisioned IOPS per month
Throughput Optimized HDD	\$0.045 per GB per month
Cold HDD	\$0.025 per GB per month
Snapshots	\$0.05 per GB per month

Keep in mind that you are still billed for EBS volumes even if they are disconnected from any EC2 instance. An excellent cost-saving strategy in this case is to take a snapshot of the volume and then delete it so that you no longer incur the charges for EBS.

EFS SERVICES

Once again, with EFS you pay for only what you use. With this service, there are no additional costs for bandwidth or requests when using the default mode.

Here are some additional facts regarding EFS service costs:



- There are additional charges for EFS File Sync. This is a pay-as-you-go

model for data copied to EFS on a per-gigabyte basis. You can easily track this service with Amazon CloudWatch. The US West (Oregon) EFS File Sync price per month per gigabyte is \$0.01.

- ■ In the default EFS Bursting Throughput mode, there is no charge for transfers as described previously given a baseline rate of 50 KB/s per gigabyte of throughput. The cost of EFS in US West (Oregon) given this default mode is \$0.30 per gigabyte per month.
- ■ There is an optional EFS Provisioned Throughput mode. You are charged only above the baseline rate of 50 KB/s per gigabyte of throughput. The charge in US West (Oregon) is \$6.00 per MB/s per month.
- ■ EFS can be perceived as being much more expensive than EBS. For example, EBS typically costs around \$0.10/GB while EFS would be \$0.30/GB. In reality, however, you save money due to no overprovisioning (you pay only for actual usage). Also, if you have a replicated HA/FT environment, you would need multiple EC2 instances with multiple attached EBS volumes. This will typically be far more expensive than using EFS. For example, consider a three-web server farm, with three Availability Zones (AZs; each with EBS volumes attached). Assuming full replication among all three instances, you would NET \$0.30/GB of stored data anyhow—the cost of EC2 processing, cross-AZ transfer fees, and additional overprovisioned storage in EBS. This makes EFS far cheaper.

GLACIER SERVICES

AWS Glacier is (currently) priced from just \$0.004 per gigabyte per month for many Regions. Upload requests are priced from \$0.05 per 1000 requests. Note that archives in Glacier have a minimum of 90 days of storage. Archives deleted before the 90 days are charged at a pro-rated charge equal to the remaining days. Like S3, storage costs for Glacier do vary by Region. Also, like Glacier, there are no charges for data transfers under certain scenarios. For example, there are no data transfer charges when transferring between EC2 and Glacier in the same Region.

You can retrieve storage from Glacier in three ways. These

different methods, described next, incur different fees:

Key Topic

- ■ **Expedited retrieval:** This cost is \$0.03 per gigabyte and \$0.01 per request.
- ■ **Standard retrieval:** This cost is \$0.01 per gigabyte and \$0.05 per 1000 requests.
- ■ **Bulk retrieval:** This cost is \$0.0025 per gigabyte and \$0.025 per 1000 requests.

Lab: Changing the Retrieval Rate in Glacier

In the following steps, you change the retrieval rate property of a newly created Glacier Vault:

Step 1. Log in to the AWS Management Console and search for **Glacier**. Click the link.

Step 2. Use the Region drop-down at the top of the console to switch to the US WEST (Oregon) Region.

Step 3. Click the **Create Vault** button.

Step 4. Use the Vault Name of MyVault. Click **Next Step**.

Step 5. Choose **Next Step** to accept the default Notifications settings.

Step 6. Choose **Submit**.

Step 7. Select your Vault and choose **Settings**.

Step 8. Set the Max Retrieval Rate to **2 GB per hour**.

Figure 13-3 shows this screen.

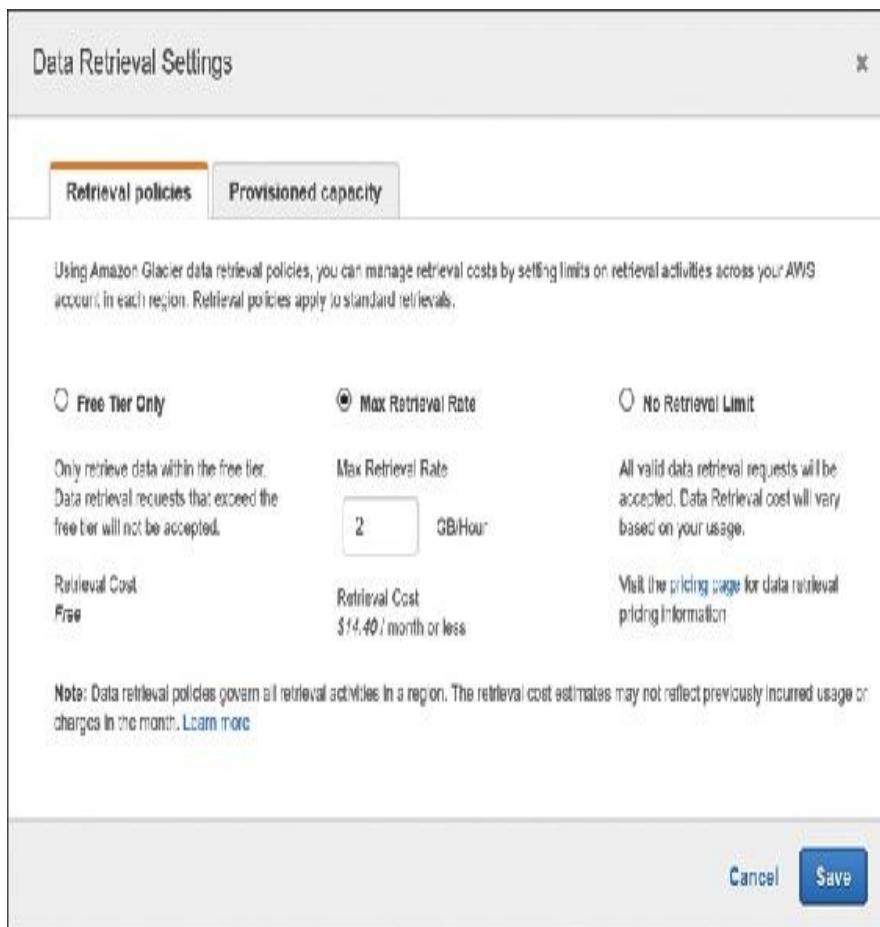


Figure 13-3 Configuring the Max Retrieval Rate

Step 9. Choose Save.

Lab Cleanup

Step 1. In the Amazon Glacier Vaults page of the AWS Management Console, select the Vault you created in this lab.

Step 2. Click **Delete Vault**. Then click **Delete Vault** again.

EXAM PREPARATION TASKS

As mentioned in the section “How to Use This Book” in the Introduction, you have a couple of choices for exam preparation: the exercises here, Chapter 16, “Final Preparation,” and the exam simulation questions in the Pearson Test Prep practice test software.

REVIEW ALL KEY TOPICS

Review the most important topics in this chapter, noted with the Key Topics icon in the margin of the page. Table 13-3 lists a reference of these key topics and the page numbers on which each is found.

Table 13-3 Key Topics for Chapter 13

		
Key Topic Element	Description	Page Number
Steps	Estimating AWS S3 Costs	214
List	Facts regarding EFS charges	219
List	Glacier data retrieval options	219

COMPLETE TABLES AND LISTS FROM MEMORY

MEMORY

There are no memory tables in this chapter.

DEFINE KEY TERMS

Define the following key terms from this chapter and check your answers in the glossary:

AWS Simple Monthly Calculator

Versioning

Archive

Q&A

The answers to these questions appear in Appendix A. For more practice with exam format questions, use the Pearson Test Prep practice test software.

1. Name at least three metrics for charges in S3.
2. What would be the most expensive storage type for EBS?
3. What is a tool for EFS file transfer?
4. What are the three different Glacier retrieval tiers?

Chapter 14. Cost-Optimized Compute

This chapter covers the following subjects:

- **Cost-Optimized EC2 Services:** EC2 is often the backbone of your AWS infrastructure. Therefore, understanding ways to control costs is critical. This section deals with this issue head-on.
- **Cost-Optimized Lambda Services:** Serverless computing power is quickly becoming the rage. This section discusses cost optimization in AWS Lambda.

Ensuring that your costs are under control in AWS often deals with your computing horsepower. You might be using EC2 virtual machine resources, or you might be relying on the serverless compute power you are using with AWS Lambda. This chapter analyzes both cases with an eye toward saving costs with each.

“DO I KNOW THIS ALREADY?” QUIZ

The “Do I Know This Already?” quiz allows you to assess whether you should read the entire chapter. Table 14-1 lists the major headings in this chapter and the “Do I Know This Already?” quiz questions covering the material in those headings so you can assess your knowledge of these specific areas. The answers to the “Do I Know This Already?” quiz appear in Appendix A.

Table 14-1 “Do I Know This Already?” Foundation Topics
Section-to-Question Mapping

Foundation Topics Section	Question
Cost-Optimized EC2 Services	1–2

Cost-Optimized Lambda Services 3–4

Caution

The goal of self-assessment is to gauge your mastery of the topics in this chapter. If you do not know the answer to a question or are only partially sure of the answer, you should mark that question as wrong for purposes of the self-assessment. Giving yourself credit for an answer you correctly guess skews your self-assessment results and might provide you with a false sense of security.

- 1.** Which of the following is not an EC2 instance category?
 - a.** Compute Optimized
 - b.** General Purpose
 - c.** Cost Optimized
 - d.** Storage Optimized
- 2.** Which of the following is not a type of reserved instance in AWS?
 - a.** Convertible
 - b.** Standard
 - c.** Scheduled
 - d.** Spot
- 3.** You have learned that your Lambda billing is based on

function duration. What is this duration rounded to for billing purposes?

a. 2 seconds

b. 1 second

c. 10 ms

d. 100 ms

4. You would like to manipulate your Lambda settings so that a key function completes in under 80 ms. What setting do you manipulate to achieve this?

a. Manipulate the memory assignment.

b. Increase the number of CPUs dedicated to the function.

c. Increase the caching ratio for the function.

d. Increase the maximum EC2 instance parameter.

FOUNDATION TOPICS

COST-OPTIMIZED EC2 SERVICES

Because EC2 makes up the foundation of so many AWS architectures, it is important to consider costs in the design.

Following are some critical considerations for cost-optimized EC2 approaches:

- **Choose the correct instance size for your workload:** Costs associated with your EC2 deployment have the advantage of scaling up resources only as you need them. You can save on costs initially by using lower-cost, lower-powered alternatives. Remember, your choice of instance type drives costs as well as the computing capacity you may use. The following categories of instances are available to you at varying price points:

- ■ General Purpose
- ■ Compute Optimized
- ■ Memory Optimized
- ■ Accelerated Computing
- ■ Storage Optimized
- ■ **Save costs through the use of reserved instances:** Using reserved computing capacity in AWS can save as much as 75 percent off using the default on-demand pricing model. There are three different types of reserved instances:
 - ■ **Standard RIs:** These RIs are best suited for steady-state usage. These types offer the largest potential discount over on-demand computing instances.
 - ■ **Convertible RIs:** These RIs enable you to change the attributes of the RIs. Your changes must be of equal or lesser cost value to the initial reservation. The potential discount with these types is lower, reaching approximately 55 percent off on-demand pricing.
 - ■ **Scheduled RIs:** These RIs are launched within a certain time window.

Note

A common misconception is that an RI is a contract on an instance. Instead, it is merely instance capacity. For example, if you run 100 EC2 instances, but many come and go at any given time, and you purchase 20 RIs, those RIs are applied to the capacity, evaluated every hour. So every hour the billing system indicates you have 20 RIs to apply, and it looks for capacity to apply the contract pricing to. It is not bound to an instance.

- ■ **Consider the use of the spot market for EC2 instances:** This approach enables you to bid on EC2 compute capacity. Most of the time spot pricing is substantially cheaper than on-demand and many times cheaper than RIs. The price is based on supply and demand at the AZ level. Savings can be as much as 90 percent off other alternatives.
- ■ **Monitor, track, and analyze service usage:** You can use CloudWatch and Trusted Advisor to ensure you are right-sizing your EC2 computing

power and balancing it against costs.

- **Analyze costs with tools in your billing dashboard and the Cost Explorer online tool:**

You can set billing alarms, budgets, and other such valuable monitoring tools within the billing dashboard. You can also use powerful cost estimators when needed. Online you can find the Simple Monthly Cost Estimator that can help you with architecting decisions around costs.

Lab: Using the Cost Explorer

To engage in correct cost optimization, you must understand how you can analyze your existing costs in the AWS system. In the following steps, you use the Cost Explorer:

Step 1. Log in to the AWS Management Console using your root credentials.

Step 2. Select the drop-down menu under your account name at the top of the console, and then choose **My Billing Dashboard**.

Step 3. From the navigation links on the left side, choose **Cost Explorer**, as shown in Figure 14-1.

Step 4. Select the **Launch Cost Explorer** link.

Step 5. From the Reports drop-down menu, choose **Daily Costs**. The report is displayed, as shown in Figure 14-2.

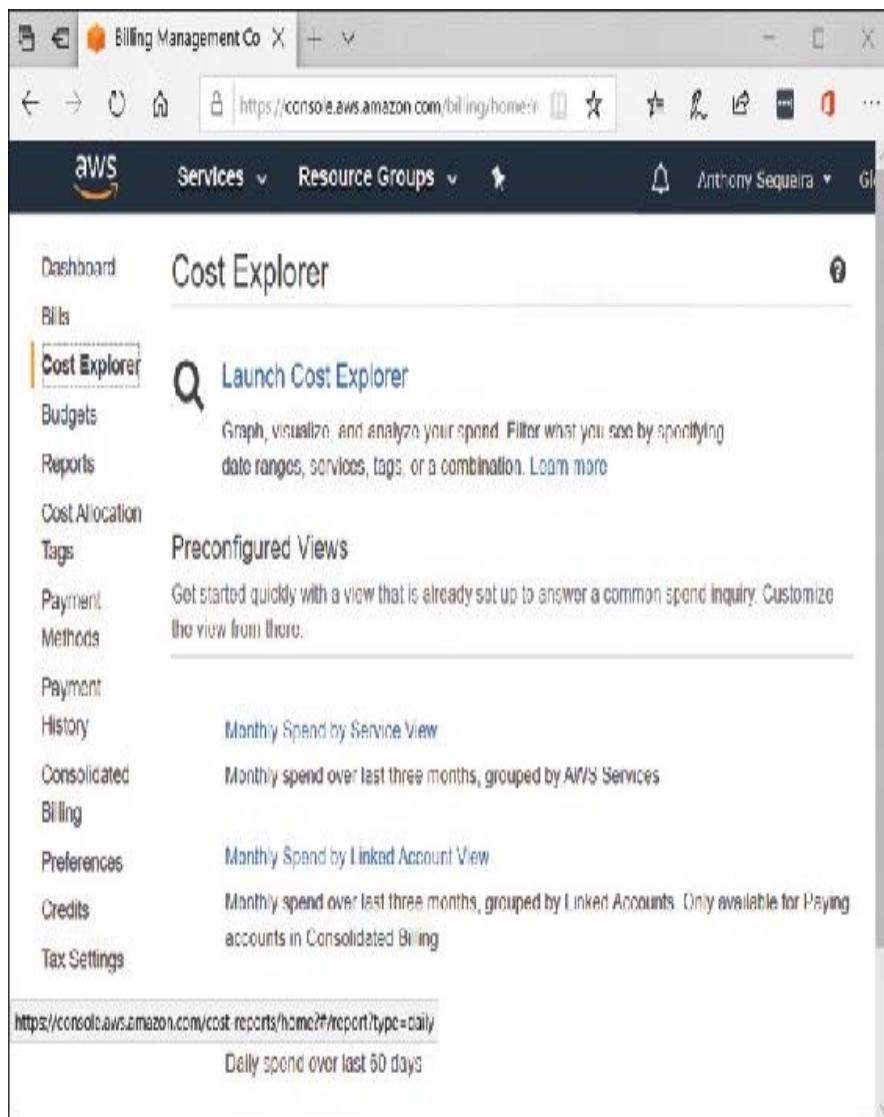


Figure 14-1 The AWS Cost Explorer



Figure 14-2 The Daily Costs report in the Cost Explorer

Step 6. Select the **Date Range** drop-down menu. Under Historical, near the bottom, choose **1M** and click **Apply**. Note that your chart is updated to reflect the new date range.

Lab: Creating a Billing Alarm

Thanks to CloudWatch, you can easily create alarms or alerts based on cost information if costs exceed a certain threshold.

To do so, follow these steps:

Step 1. Log in to the AWS console using your root credentials.

Step 2. Search your AWS services for CloudWatch and select the link.

Step 3. Select **Alarms** from the navigation links on the left and click the **Create Alarm** button.

Step 4. In the CloudWatch Metrics by Category window, click the **Total Estimated Charge** link under Billing Metrics, as shown in Figure 14-3.

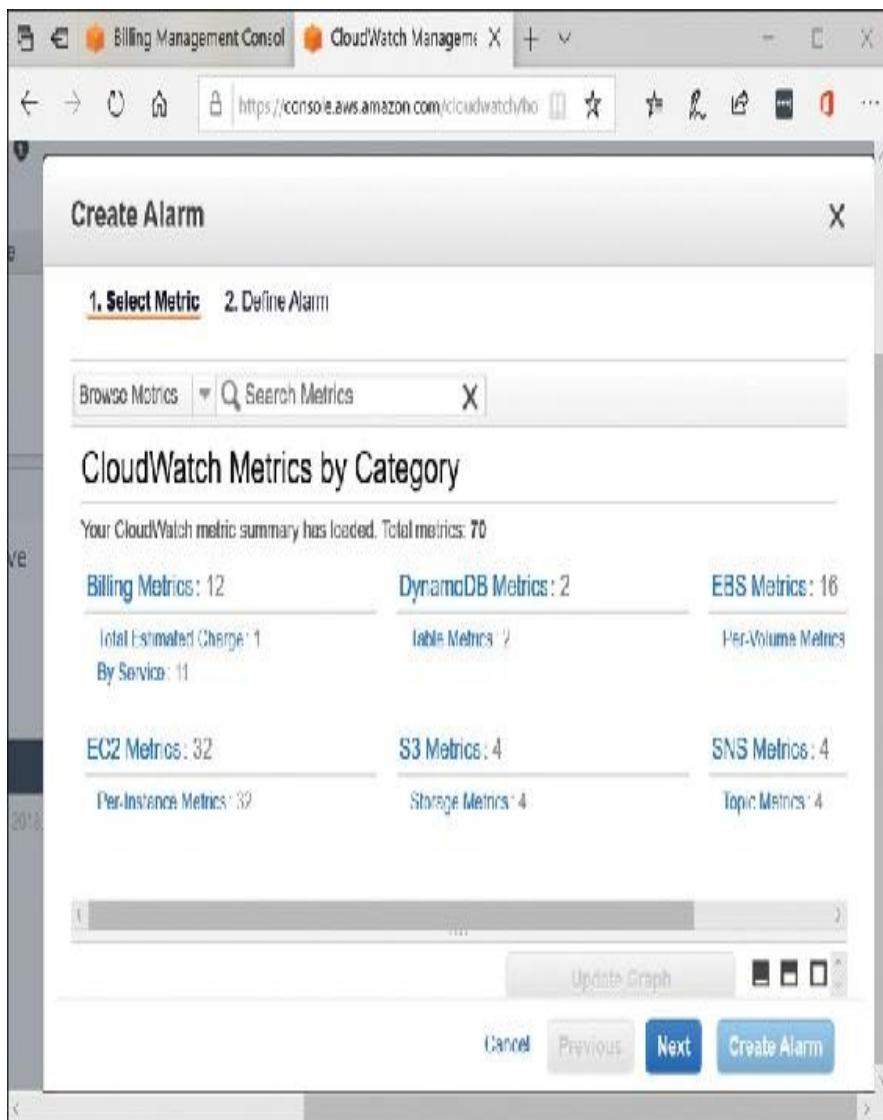


Figure 14-3 Configuring an Alarm Based on Total Estimated Charges

Step 5. Select **USD** currency for the Estimated Charges Metric Name and click **Next**.

Step 6. Name the Alarm **My Billing Alarm** and set a Greater Than or Equal To Threshold of \$30.

Step 7. Click **New List** under Actions to create a new notification list. Enter a topic name **MyNewList** and enter your email address in the Email List field.

Click **Create Alarm** when finished.

Step 8. Choose **I Will Do It Later** in the Confirm New Email Addresses window. Your new alarm is complete.

COST-OPTIMIZED LAMBDA SERVICES

Serverless computing with AWS Lambda is exciting for many reasons. One of them relates directly to costs. The reason is that you are able to more accurately follow a pay-per-use model. You are not attempting to right-size EC2 instances; in fact, you are not provisioning EC2 instances at all. Instead, you are running code against Lambda and being billed for the time the code is running. Lambda takes care of the scaling when such scaling is required.

Note the following regarding Lambda before delving deeper into cost optimization:

- ■ You can choose the amount of memory available for functions you are to run; this assignment ranges from 128 MB to 3 GB.
- ■ Based on memory selection, Lambda allocates a proportional amount of CPU and other resources.
- ■ Billing is based on gigabytes per seconds consumed; this means that a 256 MB function invocation that runs for 100 ms will cost twice as much as a 128 MB function that runs for 100 ms.
- ■ For billing purposes, function duration is rounded to the nearest 100 ms.

You should begin cost optimization around Lambda by asking yourself some important questions:

- ■ How much of your total compute costs are Lambda? You might find that a small percentage of costs associated with AWS are actually Lambda. You might be better served optimizing costs with the surrounding resources such as databases and required EC2 instances.

- ■ What are the performance requirements of the solution you are architecting? You must be careful not to cause performance degradation issues based on your cost optimizations.
- ■ What are the Lambda functions that are most critical and run the most frequently? Optimizing the costs of these functions will, of course, have the greatest impact. Costs associated with infrequently run functions are not of concern because they will cost relatively little.

For Lambda, two primary metrics are of concern when cost optimizing:

- ■ **Allocated Memory Utilization:** Each time a Lambda function is invoked, two memory-related values are printed to CloudWatch logs. They are Memory Size and Max Memory Used. Memory Size is the function's memory setting. Max Memory Used is how much memory was actually used during function invocation. Watching this metric, you can decrease memory allocation on functions that are overprovisioned and watch for increasing memory use that may indicate functions becoming underallocated.
- ■ **Billed Duration Utilization:** Remember that Lambda usage is billed in 100 ms intervals. Like memory usage, Duration and Billed Duration are logged in to CloudWatch after each function invocation. You can use these values to calculate a metric representing the percentage of billed time for which your functions were running. Although 100 ms billing intervals are granular compared to most pay-to-provision services, there can still be major cost implications to watch out for. Consider a 1 GB function that generally runs in 10 ms. Each invocation of this function will be billed as if it takes 100 ms, a 10× difference in cost! In this case it may make sense to decrease the memory setting of this function so that the runtime is closer to 100 ms with significantly lower costs. An alternative approach is to rewrite the function to perform more work per invocation—for example, processing multiple items from a queue instead of one—to increase Billed Duration Utilization. Conversely, in some cases, increasing the memory setting can result in lower costs and better performance. Consider a 1 GB function that runs in 110 ms. This will be billed as 200 ms. Increasing the memory setting slightly may allow the function to execute under 100 ms, which will decrease the billing duration by 50 percent and result in lower costs. Also realize that a “hung” function that has a 5-minute timeout set and normally runs for 100 ms equates to a 3000× charge. Carefully managing timeouts, failing fast/early, and building asynchronous stateless functions are crucial to cost management.

EXAM PREPARATION TASKS

As mentioned in the section “How to Use This Book” in the Introduction, you have a couple of choices for exam preparation: the exercises here, Chapter 16, “Final Preparation,” and the exam simulation questions in the Pearson Test Prep practice test software.

REVIEW ALL KEY TOPICS

Review the most important topics in this chapter, noted with the Key Topics icon in the margin of the page. Table 14-2 lists a reference of these key topics and the page numbers on which each is found.

Table 14-2 Key Topics for Chapter 14

Key Topic Element	Description	Page Number
		
List	EC2 cost optimization guidelines	227
List	Lambda cost optimization considerations	231

COMPLETE TABLES AND LISTS FROM

MEMORY

There are no memory tables in this chapter.

DEFINE KEY TERMS

Define the following key terms from this chapter and check your answers in the glossary:

Reserved instances

Standard RI

Convertible RI

Scheduled RI

Cost Explorer

Billing metrics

Q&A

The answers to these questions appear in Appendix A. For more practice with exam format questions, use the Pearson Test Prep practice test software.

- 1.** What are the three types of reserved instances, and which offers the greatest cost savings?

- 2.** Describe how Lambda cost charges are based.



Part V

Domain 5: Define Operationally Excellent Architectures

Chapter 15. Features for Operational Excellence

This chapter covers the following subjects:

- ■ **Introduction to the AWS Well-Architected Framework:** This section describes the Well-Architected Framework tenets as described by Amazon.
- ■ **Prepare:** This section details step-by-step guidelines for preparing the operationally excellent AWS solution.
- ■ **Operate:** This section describes key best practices for actually operating your system.
- ■ **Evolve:** This section shares tips for consistently improving your environment through small changes.

This chapter focuses on tried-and-true best practices from Amazon for creating operationally excellent solutions in AWS. As you will learn, there are three main areas: Prepare, Operate, and Evolve. The chapter breaks down each of these areas into best practices that should make a measurable difference in the functionality and success of your solutions.

“DO I KNOW THIS ALREADY?” QUIZ

The “Do I Know This Already?” quiz allows you to assess whether you should read the entire chapter. Table 15-1 lists the major headings in this chapter and the “Do I Know This Already?” quiz questions covering the material in those headings so you can assess your knowledge of these specific areas. The answers to the “Do I Know This Already?” quiz appear in Appendix A.

Table 15-1 “Do I Know This Already?” Foundation Topics

Section-to-Question Mapping

Foundation Topics Section	Question
Introduction to the AWS Well-Architected Framework	1–2
Prepare	3
Operate	4
Evolve	5

Caution

The goal of self-assessment is to gauge your mastery of the topics in this chapter. If you do not know the answer to a question or are only partially sure of the answer, you should mark that question as wrong for purposes of the self-assessment. Giving yourself credit for an answer you correctly guess skews your self-assessment results and might provide you with a false sense of security.

- 1.** Which is not a pillar in the AWS Well-Architected Framework?
 - a.** Reliability
 - b.** Orchestration
 - c.** Cost Optimization

d. Security

2. When designing an AWS solution, how should you perform as many operations as possible?

- a.** With code
- b.** Using migrations from on-premises
- c.** Using the root account
- d.** Using a single AZ

3. What tool analyzes your architecture and makes best practice recommendations?

- a.** Cloud Compliance
- b.** CloudTrail
- c.** CloudWatch
- d.** Trusted Advisor

4. Which are considered best practices in the Operate phase of operational excellence? (Choose two.)

- a.** Understanding operational health
- b.** Learning from experience
- c.** Sharing learnings
- d.** Responding to events

5. What tool would you use to track API calls to your AWS infrastructure?

- a.** GitHub
- b.** CloudTrail
- c.** CloudWatch
- d.** Trusted Advisor

FOUNDATION TOPICS

INTRODUCTION TO THE AWS WELL-ARCHITECTED FRAMEWORK

Figure 15-1 shows the AWS Well-Architected Framework.



Figure 15-1 The AWS Well-Architected Framework

This important set of architectural guidelines helps you

construct best practice designs with the following in mind:

- ■ Reliability
- ■ Security
- ■ Efficiency
- ■ Cost effectiveness

Following the preceding goals, the complete framework consists of the following pillars:

- ■ Operational Excellence
- ■ Security
- ■ Reliability
- ■ Performance Efficiency
- ■ Cost Optimization

This focus of this chapter is on the first pillar—Operational Excellence. For coverage of the other pillars, search Google on the keywords *AWS Well Architected*.

Before we delve deep into the three main parts of the Operational Excellence pillar (Prepare, Operate, and Evolve), let's examine the six design principles for operational excellence in the cloud:



- ■ **Perform operations as code:** Attempt to code as much of your AWS implementation and operation as possible. Remember that your entire architecture and operations should be “code-able”; this reduces or even eliminates the human error factor.
- ■ **Create annotated documentation:** Automate the creation of annotated documentation of your environment.
- ■ **Make changes small and frequent and reversible:** Design your system so that regular updates are possible, ensure that changes can be small in

scope, and ensure that changes can easily be reversed in the event of disaster.

- **■ Refine operations procedures frequently:** Observe ongoing operations closely and design improvements to evolve the overall system.
- **■ Anticipate failures:** Be proactive against issues that might occur in your design. Test failure scenarios.
- **■ Learn from all operational failures:** Share what is learned from any failures.

To ensure that the AWS Well-Architected Framework is comprehensive in the area of operational excellence, Amazon breaks it down into three main areas: Prepare, Operate, and Evolve (see [Figure 15-2](#)). The following sections of this chapter cover these areas in detail.

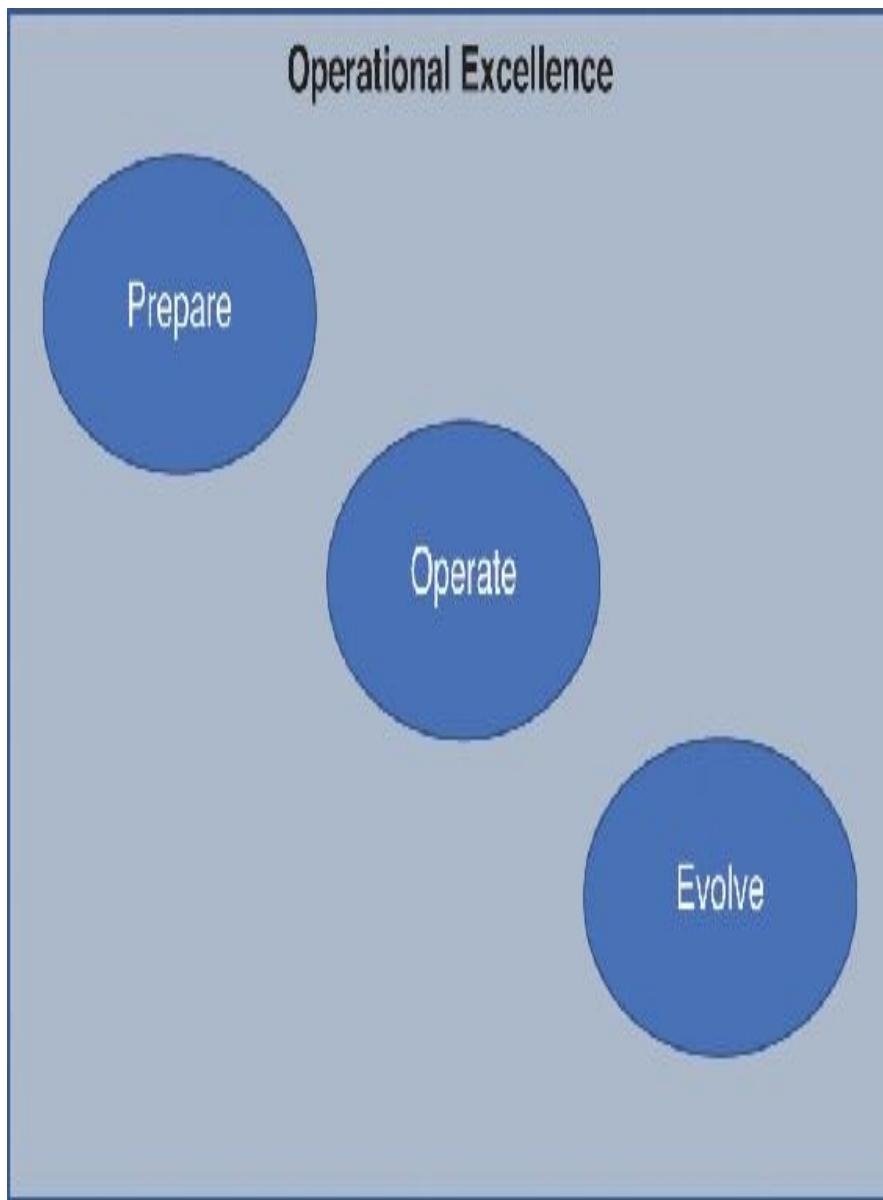


Figure 15-2 The Three Main Areas of Focus for Operational Excellence

PREPARE

The key to the Prepare area of operational excellence is for you to fully understand your AWS workloads and the behaviors you expect from the design. To be fully prepared for eventual operational excellence, you should focus on three

main areas: operational priorities, design for operations, and operational readiness.

Operational Priorities

It is important to analyze the business objectives and goals of the workload and use them to set operational priorities inside the AWS design. Keep in mind that regulatory and compliance requirements might affect your priorities. It is important to set the operational priorities so that you can make improvements in the AWS design that will have the greatest impact on the business objectives.

Fortunately, services and features in AWS can assist in this area. They include

- **AWS Cloud Compliance:** Provides valuable information about achieving the highest levels of compliance with various security requirements.
- **AWS Trusted Advisor:** Provides real-time guidance on your AWS operations and helps to ensure you are following best practices. Business Support allows full access to the full set of Trusted Advisor checks.
- **Enterprise Support:** Provides access to Technical Account Managers (TAMs) that can provide guidance on operational excellence.

Design for Operations

Keep these facts in mind as you design for operational excellence:



- In your design, include how workloads will be deployed, updated, and operated.
- Implement engineering practices that reduce defects and allow for quick and safe fixes.
- Enable observation with logging, instrumentation, and insightful metrics.

- ■ Design for a code-based implementation and ensure you can also update using code.
- ■ Consider the use of CloudFormation for the construction of version-controlled templates for your infrastructure.
- ■ Set up a Continuous Integration/Continuous Deployment (CI/CD) pipeline using the AWS Developer Tools.
- ■ Apply metadata using tags to help identify resources related to operational activities.
- ■ Design CloudWatch into your system—capture logs, inspect logs, use events.
- ■ Whenever possible, code applications to share metric information with CloudWatch.
- ■ Trace the execution of distributed applications using [AWS X-Ray](#).
- ■ When adding instrumentation to workloads, capture a broad set of information to maintain situational awareness.

Operational Readiness

Follow these guidelines in the operational readiness area to help ensure operational excellence:

- ■ Use tools like detailed checklists to know when your workloads are ready for production; use a governance process to make informed decisions regarding launches.
- ■ Use runbooks that document routine activities.
- ■ Use playbooks that guide processes for issue resolution.
- ■ Ensure enough team members are on staff for operational needs.
- ■ Use as many scripted procedures as possible to foster automation.
- ■ Consider the automatic triggering of scripted procedures based on events.
- ■ Consider scripting routines in the consistent evaluation of your AWS architecture.
- ■ Test failure procedures and the success of your responses.
- ■ Consider parallel environments for testing purposes.

- ■ Use scripting tools and related components like the AWS Systems Manager Run Command, Systems Manager Automation, and Lambda.
- ■ Consider the use of AWS Config to help create baselines and then test your configurations through the use of AWS Config rules.
- ■ Encourage team members to constantly further their education on AWS; they can use books like this one and the many free AWS resources found online via the AWS site.

OPERATE

As discussed previously, when you operate your architecture, you must be ready to prove success using key metrics you have identified for the project. The two key areas for operating with operational excellence are understanding operational health and responding to events.

Understanding Operational Health

It should be a simple matter for you and your staff to quickly identify the operational health of your system. Remember these points:

- ■ Track metrics to specific operational business goals.
- ■ Take advantage of the ease with which you can gather and analyze log files. Once again, automate as much as possible through code.
- ■ Create baselines using CloudWatch.
- ■ Use CloudWatch dashboards to create custom views that present system-level and business-level views of metrics.
- ■ Consider the use of Elasticsearch to create dashboards and visualizations of operational health.
- ■ Use the Service Health dashboard to watch for alerts.
- ■ Use the support for third-party log analysis systems such as Grafana, Kibana, and Logstash.

Responding to Events

Keep these best practices in mind when responding to events:

- ■ Anticipate for your planned and unplanned events.
- ■ Utilize your runbooks and playbooks.
- ■ Use CloudWatch rules to automatically trigger responses.
- ■ Consider the use of third-party tools for monitoring and automating responses. Some examples are New Relic, Splunk, Loggly, SumoLogic, and Datalog.
- ■ Know when human responses and decisions are needed.

EVOLVE

You should strive to engage in a continuous cycle of improvement of your AWS architecture over time. Increment small and frequent changes as discussed earlier in this chapter. To properly evolve your systems over time, consider learning from your experiences and sharing what you've learned.

Learning from Experience

Consider the following best practices in this area:

- ■ Provide time for the analysis of operations.
- ■ Aggregate and analyze logs.
- ■ Use temporary duplicates of environments for testing when needed.
- ■ Use CloudTrail to track API activity.
- ■ Perform cross-team reviews.

Share Learnings

You should make sure to share what you learn with other teams as appropriate. What you learn can help other teams avoid problems and respond quickly to operational concerns. Coding as much as possible makes sharing best practices that much easier. Be sure to use IAM to enact the appropriate

permissions on shared resources. You should also consider the use of third-party tools like GitHub, BitBucket, and SourceForge.

EXAM PREPARATION TASKS

As mentioned in the section “How to Use This Book” in the Introduction, you have a couple of choices for exam preparation: the exercises here, Chapter 16, “Final Preparation,” and the exam simulation questions in the Pearson Test Prep practice test software.

REVIEW ALL KEY TOPICS

Review the most important topics in this chapter, noted with the Key Topics icon in the margin of the page. Table 15-2 lists a reference of these key topics and the page numbers on which each is found.

Table 15-2 Key Topics for Chapter 15

Key Topic		
Key Topic Element	Description	Page Number
List	Design best practices	238
List	Designing for operational excellence best	239

practices

COMPLETE TABLES AND LISTS FROM MEMORY

There are no memory tables in this chapter.

DEFINE KEY TERMS

Define the following key terms from this chapter and check your answers in the glossary:

AWS Cloud Compliance

Technical Account Managers (TAMs)

AWS X-Ray

AWS Config

Q&A

The answers to these questions appear in Appendix A. For more practice with exam format questions, use the Pearson Test Prep practice test software.

1. What are the pillars of the AWS Well-Architected Framework?
2. Name at least three design principles for operational excellence.
3. What are the three main areas of operational excellence?

Part VI

Final Preparation

Chapter 16. Final Preparation

Are you excited for your exam after reading this book? I sure hope so. You should be. In this chapter we put certification prep all together for you. This includes taking a more detailed look at the actual certification exam itself.

This chapter shares some great ideas on ensuring you ace that exam. If you read this book with the interest of really mastering AWS and you were not really considering certification, maybe this chapter will convince you to give it a try!

The first 15 chapters of this book cover the technologies, protocols, design concepts, and considerations required to be prepared to pass the AWS Certified Solutions Architect - Associate (SAA-C01) exam. Although these chapters supply the detailed information, most people need more preparation than just reading the first 15 chapters of this book. This chapter details a set of tools and a study plan to help you complete your preparation for the exams.

This short chapter has four main sections. The first section lists the AWS Certified Solutions Architect - Associate (SAA-C01) exam information and breakdown. The second section shares some important tips to keep in mind to ensure you are ready for this exam. The third section discusses exam preparation tools useful at this point in the study process. The final section of this chapter lists a suggested study plan now that you have completed all the earlier chapters in this book.

Note

Note that [Appendix C, “Memory Tables,”](#) and [Appendix D, “Memory Tables Answer Key,”](#) exist as soft-copy appendixes on the website for this book, which you can access by going to www.pearsonITcertification.com/register, registering your book, and entering this book’s ISBN: 9780789760494.

EXAM INFORMATION

Here are details you should be aware of regarding the exam that maps to this text.

Question Types: Multiple Choice and Multiple Correct Answer Style Multiple Choice

Number of Questions: 65

Time Limit: 130 minutes

Required Passing Score: 720 out of 1000

Available Languages: English, Japanese, Simplified Chinese, Korean

Exam Fee: 150 USD

Exam ID Code: SAA-C01

This exam seeks to validate the following for a candidate:

- Define a solution using architectural design principles based on customer requirements.
- Provide implementation guidance based on best practices to the organization throughout the lifecycle of the project.

Amazon Web Service certification exam authors recommend the following skills and experience for candidates wanting to pass this exam:

- One year of hands-on experience designing available, cost-efficient, fault-tolerant, and scalable distributed systems on AWS

- ■ Hands-on experience using compute, networking, storage, and database AWS services
- ■ Hands-on experience with AWS deployment and management services
- ■ Ability to identify and define technical requirements for an AWS-based application
- ■ Ability to identify which AWS services meet a given technical requirement
- ■ Knowledge of recommended best practices for building secure and reliable applications on the AWS platform
- ■ An understanding of the basic architectural principles of building on the AWS cloud
- ■ An understanding of the AWS global infrastructure
- ■ An understanding of network technologies as they relate to AWS
- ■ An understanding of security features and tools that AWS provides and how they relate to traditional services

The exam is broken up into five different domains. Here are those domains and the percentage of the exam for each of the domains:

- ■ Design Resilient Architectures: 34 percent
- ■ Design Performant Architectures: 24 percent
- ■ Specify Secure Applications and Architectures: 26 percent
- ■ Design Cost-Optimized Architectures: 10 percent
- ■ Define Operationally Excellent Architectures: 6 percent

Here is the breakdown of the exact exam objectives for these various domains:

Domain 1: Design Resilient Architectures

1.1 Choose reliable/resilient storage.

1.2 Determine how to design decoupling mechanisms using AWS services.

1.3 Determine how to design a multitier architecture solution.

1.4 Determine how to design high availability and/or fault-tolerant architectures.

Domain 2: Define Performant Architectures

2.1 Choose performant storage and databases.

2.2 Apply caching to improve performance.

2.3 Design solutions for elasticity and scalability.

Domain 3: Specify Secure Applications and Architectures

3.1 Determine how to secure application tiers.

3.2 Determine how to secure data.

3.3 Define the networking infrastructure for a single VPC application.

Domain 4: Design Cost-Optimized Architectures

4.1 Determine how to design cost-optimized storage.

4.2 Determine how to design cost-optimized compute.

Domain 5: Define Operationally Excellent Architectures

5.1 Choose design features in solutions that enable operational excellence.

GETTING READY

Here are some important tips to keep in mind to ensure you are ready for this rewarding exam!

- **Build and use a study tracker:** Consider taking the exam objectives shown in this chapter and build yourself a study tracker! This will help ensure you have not missed anything and that you are confident for your

exam! As a matter of fact, this book offers a sample Study Planner as a website supplement.

- **Think about your time budget for questions in the exam:** When you do the math, you realize that you have 2 minutes per question. Although this does not sound like enough time, realize that many of the questions will be very straightforward, and you will take 15 to 30 seconds on those. This builds time for other questions as you take your exam.
- **Watch the clock:** Check in on the time remaining periodically as you are taking the exam. You might even find that you can slow down pretty dramatically as you have built up a nice block of extra time.
- **Get some ear plugs:** The testing center might provide ear plugs, but get some just in case and bring them along. There might be other test takers in the center with you, and you do not want to be distracted by their screams. I personally have no issue blocking out the sounds around me, so I never worry about this, but I know it is an issue for some.
- **Plan your travel time:** Give yourself extra time to find the center and get checked in. Be sure to arrive early. As you test more at that center, you can certainly start cutting it closer time-wise.
- **Get rest:** Most students report success with getting plenty of rest the night before the exam. All-night cram sessions are not typically successful.
- **Bring in valuables but get ready to lock them up:** The testing center will take your phone, your smart watch, your wallet, and other such items. It will provide a secure place for them.
- **Take notes:** You will be given note-taking implements, so do not be afraid to use them. One thing I always end up jotting down is any questions I struggled with. I then memorize these at the end of the test by reading my notes over and over again. I then always make sure I have a pen and paper in the car. I write down the issues in there just after the exam. When I get home with a pass or fail, I research those items!
- **Use the FAQs in your study:** The Amazon test authors have told me they love to pull questions from the FAQs they publish at the AWS site. These are a really fun and valuable read anyway, so go through them for the various services that are key for this exam.
- **Practice exam questions are great—use them:** This text provides many practice exam questions. Be sure to go through them thoroughly. Remember, just don't blindly memorize answers; let the questions really demonstrate

where you are weak in your knowledge and then study up on those areas.

TOOLS FOR FINAL PREPARATION

This section lists some information about the available tools and how to access them.

Pearson Test Prep Practice Test Engine and Questions on the Website

Register this book to get access to the Pearson Test Prep practice test engine (software that displays and grades a set of exam-realistic multiple-choice questions). Using the Pearson Test Prep practice test engine, you can either study by going through the questions in Study mode or take a simulated (timed) AWS Certified Solutions Architect - Associate exam.

The Pearson Test Prep practice test software comes with two full practice exams. These practice tests are available to you either online or as an offline Windows application. To access the practice exams that were developed with this book, see the instructions in the card inserted in the sleeve in the back of the book. This card includes a unique access code that enables you to activate your exams in the Pearson Test Prep software.

Accessing the Pearson Test Prep Practice Test Software Online

The online version of this software can be used on any device with a browser and connectivity to the Internet, including desktop machines, tablets, and smartphones. To start using your practice exams online, follow these steps:

Step 1. Go to <http://www.PearsonTestPrep.com>.

Step 2. Select **Pearson IT Certification** as your product group.

Step 3. Enter your email/password for your account. If you don't have an account on PearsonITCertification.com or CiscoPress.com, you will need to establish one by going to PearsonITCertification.com/join.

Step 4. In the My Products tab, click the **Activate New Product** button.

Step 5. Enter the access code printed on the insert card in the back of your book to activate your product.

Step 6. The product will now be listed in your My Products page. Click the **Exams** button to launch the exam settings screen and start your exam.

Accessing the Pearson Test Prep Practice Test Software Offline

If you want to study offline, you can download and install the Windows version of the Pearson Test Prep software. You can find a download link for this software on the book's companion website, or you can just enter this link in your browser:

[http://www.pearsonitcertification.com/content/downloads/p
cpt/engine.zip](http://www.pearsonitcertification.com/content/downloads/p cpt/engine.zip)

To access the book's companion website and the software, follow these steps:

Step 1. Register your book by going to PearsonITCertification.com/register and entering the ISBN: **9780789760494**.

Step 2. Respond to the challenge questions.

Step 3. Go to your account page and select the **Registered**

Products tab.

Step 4. Click on the **Access Bonus Content** link under the product listing.

Step 5. Click the **Install Pearson Test Prep Desktop Version** link under the Practice Exams section of the page to download the software.

Step 6. After the software finishes downloading, unzip all the files on your computer.

Step 7. Double-click the application file to start the installation and follow the on-screen instructions to complete the registration.

Step 8. After the installation is complete, launch the application and select the **Activate Exam** button on the My Products tab.

Step 9. Click the **Activate a Product** button in the Activate Product Wizard.

Step 10. Enter the unique access code found on the card in the back of your book and click the **Activate** button.

Step 11. Click **Next** and then the **Finish** button to download the exam data to your application.

Step 12. You can now start using the practice exams by selecting the product and clicking the **Open Exam** button to open the exam settings screen.

The offline and online versions will synch together, so saved exams and grade results recorded on one version will also be available to you on the other.

Customizing Your Exams

When you are in the exam settings screen, you can choose to take exams in one of three modes:

-  Study mode
-  Practice Exam mode
-  Flash Card mode

Study mode allows you to fully customize your exams and review answers as you are taking the exam. This is typically the mode you would use first to assess your knowledge and identify information gaps. Practice Exam mode locks certain customization options because it is presenting a realistic exam experience. Use this mode when you are preparing to test your exam readiness. Flash Card mode strips out the answers and presents you with only the question stem. This mode is great for late-stage preparation when you really want to challenge yourself to provide answers without the benefit of seeing multiple-choice options. This mode does not provide the detailed score reports that the other two modes do, so you should not use it if you are trying to identify knowledge gaps.

In addition to these three modes, you are able to select the source of your questions. You can choose to take exams that cover all the chapters, or you can narrow your selection to just a single chapter or the chapters that make up specific parts in the book. All chapters are selected by default. If you want to narrow your focus to individual chapters, simply deselect all the chapters and then select only those on which you want to focus in the Objectives area.

You can also select the exam banks on which to focus. Each exam bank comes complete with a full exam of questions that cover topics in every chapter. The two exams printed in the

book are available to you as well as two additional exams of unique questions. You can have the test engine serve up exams from all four banks or just from one individual bank by selecting the desired banks in the exam bank area.

You can make several other customizations to your exam from the exam settings screen, such as the time of the exam, the number of questions served up, whether to randomize questions and answers, whether to show the number of correct answers for multiple-answer questions, or whether to serve up only specific types of questions. You can also create custom test banks by selecting only questions that you have marked or questions on which you have added notes.

Updating Your Exams

If you are using the online version of the Pearson Test Prep software, you should always have access to the latest version of the software as well as the exam data. If you are using the Windows desktop version, every time you launch the software, it will check to see if any updates are available for your exam data and automatically download any changes that were made since the last time you used the software. This requires that you are connected to the Internet at the time you launch the software.

Sometimes, due to many factors, the exam data may not fully download when you activate your exam. If you find that figures or exhibits are missing, you may need to manually update your exams.

To update a particular exam you have already activated and downloaded, simply select the Tools tab and select the Update Products button. Again, this is only an issue with the Windows

desktop application.

If you want to check for updates to the Pearson Test Prep practice test software, Windows desktop version, simply select the Tools tab and select the Update Application button. This way, you ensure you are running the latest version of the software engine.

Premium Edition

In addition to the free practice exam provided on the website, you can purchase additional exams with expanded functionality directly from Pearson IT Certification. The Premium Edition of this title contains an additional two full practice exams and an eBook (in both PDF and ePUB format). In addition, the Premium Edition title also has remediation for each question to the specific part of the eBook that relates to that question.

Because you have purchased the print version of this title, you can purchase the Premium Edition at a deep discount. A coupon code in the book sleeve contains a one-time-use code and instructions for where you can purchase the Premium Edition.

To view the Premium Edition product page, go to
www.informit.com/title/9780789760494.

Memory Tables

Like most Cert Guides, this book purposely organizes information into tables and lists for easier study and review. Rereading these tables can be very useful before the exam. However, it is easy to skim over the tables without paying attention to every detail, especially when you remember

having seen the table's contents when reading the chapter.

Instead of just reading the tables in the various chapters, this book's Appendixes C and D give you another review tool.

Appendix C lists partially completed versions of many of the tables from the book. You can open Appendix C (a PDF available on the book website after registering) and print the appendix. For review, you can attempt to complete the tables. This exercise can help you focus on the review. It also exercises the memory connectors in your brain; and it makes you think about the information without as much information, which forces a little more contemplation about the facts.

Appendix D, also a PDF located on the book website, lists the completed tables to check yourself. You can also just refer to the tables as printed in the book.

Chapter-Ending Review Tools

Chapters 1 through 15 have several features in the “Exam Preparation Tasks” and “Q&A” sections at the end of the chapter. You might have already worked through these in each chapter. Using these tools again can also be useful as you make your final preparations for the exam.

SUGGESTED PLAN FOR FINAL REVIEW/STUDY

This section lists a suggested study plan from the point at which you finish reading through Chapter 15 until you take the AWS Certified Solutions Architect - Associate exam.

Certainly, you can ignore this plan, use it as is, or just take suggestions from it.

The plan uses these steps:

Step 1. Review key topics and DIKTA? questions: You can use the table that lists the key topics in each chapter or just flip the pages looking for key topics. Also, reviewing the Do I Know This Already? questions from the beginning of the chapter can be helpful for review.

Step 2. Complete memory tables: Open Appendix C from the book website and print the entire thing, or print the tables by major part. Then complete the tables.

Step 3. Review “Q&A” sections: Go through the Q&A questions Key Terms the end of each chapter to identify areas where you need more study.

Step 4. Use the Pearson Test Prep practice test engine to practice: You can use the Pearson Test Prep practice test engine to study using a bank of unique exam-realistic questions available only with this book.

SUMMARY

The tools and suggestions listed in this chapter have been designed with one goal in mind: to help you develop the skills required to pass the AWS Certified Solutions Architect - Associate exam. This book has been developed from the beginning to not only tell you the facts but also to help you learn how to apply the facts. No matter what your experience level leading up to taking the exams, it is our hope that the broad range of preparation tools, and even the structure of the book, helps you pass the exam with ease. We hope you do

well on the exam.

Part VII

Appendixes

Glossary

AKP Asynchronous Key Prefetch; improves Aurora performance by anticipating the rows needed to run queries in which a JOIN query requires use of the BKA Join algorithm and MRR optimization features **Amazon Redshift Advisor** An AWS tool that provides specific recommendations to increase performance and reduce costs associated with Amazon Redshift **Archive** Data that is stored in AWS Glacier

Asynchronous decoupling An approach to decoupled components in which the components do not need to be present at all times for the system to function **Aurora** A MySQL-and PostgreSQL-compatible relational database engine that combines the speed and availability of high-end commercial databases with the simplicity and cost-effectiveness of open-source databases **Auto Scaling** A feature that helps you maintain application availability and allows you to scale your Amazon EC2 capacity automatically according to conditions that you define **AWS Batch** A service for efficiently running batch jobs against AWS resources

AWS Cloud Compliance A set of online tools to assist you in achieving regulatory compliance with your AWS solutions

AWS Config A service that enables you to assess, audit, and evaluate the configurations of your AWS resources **AWS**

Global Infrastructure Regions and Availability Zones located around the globe

AWS KMS The encryption management service for AWS

AWS Simple Monthly Calculator An online tool that allows you to estimate monthly AWS charges **AWS Storage**

Gateway A service that seamlessly enables hybrid storage between on-premises storage environments and the AWS Cloud **AWS X-Ray** A tool for tracking execution of code in distributed applications

Billing metrics A set of metrics in CloudWatch that permit the careful monitoring of AWS costs; you can also set alarms based on this information **Bucket** Logical storage container in S3

Bursting Throughput Mode Throughput scales as the file system grows

CapEx Capital expenditures

ClassicLink A feature that allows you to connect EC2-Classic instances to your VPC

CloudFormation A service that gives developers and system administrators an easy way to create and manage a collection of related AWS resources, provisioning and updating them in an orderly and predictable fashion **CloudFront** A global

content delivery network (CDN) service that accelerates delivery of your websites, APIs, video content, or other web assets **CloudTrail** A web service that records AWS API calls for your account and delivers log files to you **CloudWatch** A monitoring service for AWS Cloud resources and the applications you run on AWS

Convertible RI A reserved instance that permits modifications from the original reservation **Cooldown period** Mandatory waiting period between Auto Scaling actions

Cost Explorer A tool in the Management Console that permits you to create detailed reports based on your AWS costs **Data tier** The storage media for the data required by an application

Database Migration Service A service that helps you migrate databases into or out of AWS easily and securely **DAX** DynamoDB Accelerator; a caching engine specially designed for DynamoDB

Decoupling Creating components that have the capability to be autonomous from each other to some degree **DHCP option sets** Components that allow you to provide settings such as a DNS name or DNS server address to instances **Direct Connect** A tool that makes it easy to establish a dedicated network connection from your premises to AWS

Directory Services A service that enables your directory-aware workloads and AWS resources to use managed Active Directory in the AWS Cloud **Distribution** A group of resources cached and shared by CloudFront

DNS Domain Name System; the DNS server provided by AWS or the one or more servers that you provide **Durability** Resiliency; in data storage, this term means the chance of data loss

DynamoDB A fast and flexible NoSQL database service for all applications that need consistent, single-digit millisecond latency at any scale **EBS-optimized instance** Dedicated bandwidth between EC2 and EBS

Edge location A special data center designed for delivering CloudFront cached resources **Egress-only Internet gateway**

A VPC component that provides Internet access for IPv6 addressed instances **Elastic Beanstalk** An easy-to-use service for deploying and scaling web applications and services

Elastic Block Store A resource that provides persistent block storage volumes for use with EC2 instances in the AWS Cloud

Elastic Compute Cloud (EC2) Virtual machine compute resources in AWS

Elastic Container Registry A fully-managed Docker container registry in AWS

Elastic Container Service A scalable, high-performance container management service that supports Docker containers

Elastic Container Service for Kubernetes A fully managed service that makes it easy for you to use Kubernetes on AWS without having to be an expert in managing Kubernetes

clusters **Elastic File System** A service that provides simple, scalable file storage for use with Amazon EC2 instances in the AWS Cloud **Elastic IP addresses** Public IP addresses that you can assign flexibly in your AWS VPC

Elastic Load Balancing A feature that automatically distributes incoming application traffic across multiple EC2 instances

ElastiCache A web service that makes it easy to deploy, operate, and scale an in-memory cache in the cloud

Elasticity The ability of resources to scale up and down

Fargate A technology for Amazon ECS and EKS that allows you to run containers without having to manage servers or clusters

FT Fault tolerance; a subset method of high availability designed to create a zero downtime solution and zero performance degradation due to a component failure

Glacier A secure, durable, and extremely low-cost storage

service for data archiving and long-term backup **Global secondary index** An index with a partition key and a sort key that can be different from those on the base table **Greengrass** An AWS service that allows IoT devices to run code that is synched from the cloud **Groups** Objects that contain user accounts; permissions can be assigned to groups and not users to implement scalability in the design **HA** High availability; a goal in solution design that focuses on automatic failover and recoverability from an issue in the solution **HTTPS** The secure version of HTTP for protecting data in transit

Identity and Access Management (IAM) A tool that enables you to securely control access to AWS services and resources for your users **Initialization** A process in which each block of an EBS volume is accessed in advance of deployment; it is used when you have restored a volume from a Snapshot **Instance store** Legacy ephemeral storage option for EC2 instances

Internet gateway Virtual device that provides Internet access to instances

Key Management Service (KMS) A managed service that makes it easy for you to create and control the encryption keys used to encrypt your data **Kinesis** A data platform that makes it easy to collect, process, and analyze real-time, streaming data **Lambda** A serverless compute service of AWS that permits the execution of custom code or functions without the provisioning of virtual servers **Latency** Delay (from the storage to the instance, for example)

Lazy loading Caching data strictly based on requests for the data

Lightsail An interface that enables you to launch and manage a virtual private server with AWS

Local secondary index An index that has the same partition key as the base table but a different sort key **Logic tier** The server that contains the code providing the functionality of the application or technology **Loose decoupling** Creating decoupled components that have few dependencies on each other **memcached** A simple caching engine supported by ElastiCache

Multi-AZ The reference to using multiple Availability Zones in an AWS design; often done to ensure high availability (HA) **Multi-Tier architecture** The use of multiple systems or platforms to implement an application or technology **NAT** Network Address Translation as supported in AWS

Network ACL A security mechanism in AWS used to control traffic flow in and out of subnets within VPCs **Network interface** Virtual interface that represents a NIC

NFS Network File System; a standard shared file system protocol

NIST The National Institute of Standards and Technology, which provides excellent guidance on best practices **Object storage** The data storage technique of AWS S3

OpEx Operating expenditures

OpsWorks A configuration management service that uses Chef, an automation platform that treats server configurations as code **Patch Manager** A component of AWS Systems Manager that automates the deployment of patches to operating systems in your AWS infrastructure **Policy** An

object in IAM that defines the actual permissions assigned to a resource or service in AWS

Presentation tier Components that users interact with for a user interface

Provisioned Throughput Mode An operating mode available for high throughput needs or requirements greater than

Bursting Throughput mode allows **Redis** A cache engine supported by ElastiCache that supports complex data types and multi-AZ deployments **Redshift** A fast, fully managed, petabyte-scale data warehouse that makes it simple and cost-effective to analyze all your data using your existing business intelligence tools **Relational Database Service (RDS)** A

service that makes it easy to set up, operate, and scale a relational database in the cloud **Reserved instances** AWS capacity utilized at a discount over default on-demand pricing as a result of the reservation **Role** Similar to a user but does

not represent one user in the organization; instead, it is intended to be assumable by anyone who needs it **Root user** Role created when you create your AWS account; this role has unrestricted access to all aspects of your account **Route 53** A highly available and scalable cloud Domain Name System (DNS) web service

Route table Routing instructions for your subnets

RPO Recovery point objective; the point to which data must be restored

RTO Recovery time objective; the amount of time in which a recovery must be completed **Scheduled RI** Reserved instances that will operate within a certain time window

Security group A security component in AWS used to secure EC2 and other resources

Serverless Application Repository A service that enables you to deploy code samples, components, and complete applications for common use cases such as web and mobile back ends, event and data processing, logging, monitoring, IoT, and more **Simple Notification Service (SNS)** A fast, flexible, distributed, and fully managed push notification service that lets you send individual messages or to distribute messages to many recipients **Simple Queue Service (SQS)** A fast, reliable, scalable, distributed, and fully managed message queuing service **Simple Storage Service (S3)** Object storage with a simple web service interface to store and retrieve any amount of data from anywhere on the web **Single-Tier**

Architecture A type of architecture in which all required components for a software application or technology are provided on a single server or platform **Snowball** A petabyte-scale data transport solution that uses secure appliances to transfer large amounts of data into and out of AWS

Standard RI A standard reserved instance; this type provides the greatest potential cost savings **Storage classes** Different classes of S3 storage used to balance performance and costs

Synchronous decoupling An approach to decoupling that means decoupled components must be present for the system to function properly **TCP Selective Acknowledgment** A performant S3 service that improves recovery time after large numbers of packet loss **TCP Window Scaling** A performant S3 service that supports window sizes larger than 64 KB

Technical Account Managers (TAMs) Experts that are

available to you if you have an Enterprise Support level of service **Trusted Advisor** An online resource to help you reduce cost, increase performance, and improve security by optimizing your AWS environment **TTL** The time to live value in Route 53 that dictates the amount of time resource record information cached by DNS resolvers **Users** Components in IAM that represent users in your organization

Vault Logical storage structure in AWS Glacier

Versioning Keeping multiple copies of objects in AWS S3 for various time points

Virtual Private Cloud (VPC) Virtual network components in AWS

Volume queue length The number of pending I/O requests for a device

VPC endpoints Devices that allow you to privately connect your VPC to supported endpoint services **VPC peering** A connection that permits communications between VPCs

Web Application Firewall (WAF) A feature that helps protect your web applications from common web exploits that could affect application availability, compromise security, or consume excessive resources **Write through** A caching strategy designed to minimize stale cache data

Appendix A. Answers to the “Do I Know This Already?” Quizzes and Q&A Sections

CHAPTER 1

“Do I Know This Already?” Quiz

1. d 2. a 3. c 4. d 5. c 6. c 7. a 8. c 9. d 10. c

Q&A

1. Advantages include

- ■ CapEx replaced by OpEx
- ■ Lack of contractual commitments
- ■ Reduction of required negotiations
- ■ Reduced procurement delays
- ■ “Pay as you go” model
- ■ High levels of security possible
- ■ Flexibility
- ■ A massive global infrastructure
- ■ SaaS, PaaS, and IaaS offerings
- ■ Diverse API support

2. EC2 is a major compute option in AWS. With EC2, you can easily create virtual machines that run on a hardware instance using a software definition made possible with an Amazon Machine Image (AMI).

3. S3 is object-based storage in AWS. This is an extremely scalable and flexible storage model you can use for many different purposes.
4. A Virtual Private Cloud in AWS is the collection of your virtual network resources. This includes subnets, routers, routing tables, and more.
5. The AWS Global Infrastructure consists of regions located around the world with Availability Zones inside those regions. Of course, physical data centers are located in the Availability Zones.
6. AWS offers caching, relational, nonrelational, and data warehouse database technologies.

CHAPTER 2

“Do I Know This Already?” Quiz

1. b
2. d
3. a
4. a
5. b
6. c
7. b
8. a
9. b
10. c

Q&A

1. S3 Standard, S3 Standard-IA, S3 One Zone-IA, Glacier
2. 99.99999999%
3. Three
4. The virtual machine running on the volume can be stopped and started with no data loss for the volume. EBS volumes are a separate entity and have a separate life span beyond the life of the instance. EBS volumes can be detached from one instance and attached to another. You can destroy an instance and keep EBS volumes around.

5. NFSv4

6. Backup and archiving

CHAPTER 3

“Do I Know This Already?” Quiz

1. b and d 2. d 3. a 4. b and d 5. a and d

Q&A

1. The creation of components that have the ability to be somewhat autonomous from each other.

2. To help ensure that changes to one component have a minimal impact on other components.

To help ensure that a failure in one component has a minimal impact on other components.

To promote the creation of well-defined interfaces for inter-component communication.

To enable smaller services to be consumed without prior knowledge of their network topology details.

To reduce the impact on end users when you must perform changes to the design.

3. Decoupled components that must be present in order for the system to function properly.

4. Decoupled components that do not need to be present at all times in order for the system to function.

5. The Simple Queue Service (SQS). Note that another

queue service, called AmazonMQ, is a managed Apache ActiveMQ service. SQS is far more scalable and robust, whereas AmazonMQ provides a solution for a product already written on ActiveMQ. You can think of ActiveMQ as the queuing version of ElastiCache that simply runs open-source product under the hood. At the time of this writing, AmazonMQ is not in the exam targeted by this text.

CHAPTER 4

“Do I Know This Already?” Quiz

1. c 2. c 3. b 4. c

Q&A

1. It is simple to understand and design.

It is easy to deploy.

It is easy to administer.

It has potentially lower costs associated with it.

It can be easy to test.

It is often ideal for small-scale environments that do not require scalability.

2. Improved security Improved performance

Improved scalability

Fine-grained management

Fine-grained maintenance

Higher availability

Promotes decoupling of components

Promotes multiple teams for architecture development, implementation, and maintenance

3. Presentation Tier Logic Tier

Data Tier

4. Presentation Tier—EC2

Logic Tier—Lambda

Data Tier—RDS

CHAPTER 5

“Do I Know This Already?” Quiz

1. a 2. c 3. d 4. d 5. c 6. a 7. a

Q&A

1. Elastic Load Balancing and Auto Scaling 2. Replica 3. Availability Zones

CHAPTER 6

“Do I Know This Already?” Quiz

1. a 2. a 3. b 4. a

Q&A

1. Salt the name with a random hashed prefix.

2. The AWS CLI should be considered in this case. Use the copy, move, or sync commands as needed.
3. On these volumes, network traffic has its own bandwidth compared to communications between EC2 and EBS.
4. Standard retrievals from your vault can take 3–5 hours.

CHAPTER 7

“Do I Know This Already?” Quiz

1. c
2. b
3. c
4. d
5. b
6. a
7. d
8. d
9. c
10. c

Q&A

1. Create an Amazon Aurora MySQL DB cluster and make it a replication slave of your MySQL DB instance.
2. Migrate to a DB instance class with a higher I/O capacity. Upgrade from standard storage options. Provision additional throughput capacity.
3. As a general rule, you should maintain as few tables as possible in a DynamoDB application.
4. Write through as opposed to lazy loading.
5. Advisor develops observations by running tests on your clusters to determine whether a test value is within a specific range.

CHAPTER 8

“Do I Know This Already?” Quiz

1. a and d 2. c 3. d 4. c 5. d 6. b 7. b 8. d

Q&A

1. Redis and memcached 2. A cluster 3. Edge locations 4. Lambda functions 5. DNS records

CHAPTER 9

“Do I Know This Already?” Quiz

1. a 2. b 3. c 4. c

Q&A

1. Elastic Load Balancing supports three types of load balancers: Application Load Balancers, Network Load Balancers, and Classic Load Balancers. You can select a load balancer based on your application needs.
2. No. When you delete the Auto Scaling policy, the associated CloudWatch alarms are also automatically deleted.

CHAPTER 10

“Do I Know This Already?” Quiz

1. b 2. a 3. a 4. d 5. c 6. d

Q&A

1. The three main identity components of AWS IAM are users, groups, and roles.
2. AWS Systems Manager Patch Manager.

CHAPTER 11

“Do I Know This Already?” Quiz

1. a 2. c 3. a, b, and d 4. a 5. d

Q&A

1. You control everything.

AWS controls everything.

AWS controls everything except the management areas of KMI.

CHAPTER 12

“Do I Know This Already?” Quiz

1. c 2. b 3. a 4. d 5. a 6. b 7. c 8. b and d 9. a 10. d 11. d

Q&A

1. VPC peering 2. DHCP option sets 3. Egress-only Internet gateway 4. Route tables

CHAPTER 13

“Do I Know This Already?” Quiz

1. a 2. c 3. a 4. a 5. d 6. d 7. b 8. d

Q&A

1. Storage Used Network Data Transferred In
Network Data Transferred Out

Data Requests

Data Retrieval

2. Provisioned IOPS SSD

3. EFS File Sync 4. Expedited retrieval Standard retrieval

Bulk retrieval

CHAPTER 14

“Do I Know This Already?” Quiz

1. c 2. d 3. d 4. a

Q&A

1. Standard RIs: These RIs are best suited for steady-state usage. These types offer the largest potential discount over on-demand compute instances.

Convertible RIs: These RIs enable you to change the attributes of the RIs. Your changes must be of equal or lesser cost value to the initial reservation. The potential discount with these types is lower, reaching approximately 55 percent off on-demand pricing.

Scheduled RIs: These RIs are launched within a certain time window.

2. Billing in Lambda is based on gigabytes per seconds consumed; this is based on the memory you have allocated to the function and runtime of the function rounded to the nearest 100 ms.

CHAPTER 10

“Do I Know This Already?” Quiz

- 1. b 2. a 3. d 4. a and d 5. b

Q&A

- 1. Operational Excellence Security

Reliability

Performance Efficiency

Cost Optimization

- 2. Perform operations as code.

Create automated annotated documentation.

Make small and frequent changes that are reversible.

Refine operational procedures frequently.

Anticipate failures.

Learn from all operational failures.

- 3. Prepare, Operate, Evolve

Appendix B. AWS Certified Solutions Architect – Associate (SAA-C01) Cert Guide Exam Updates

Over time, reader feedback allows Pearson to gauge which topics give our readers the most problems when taking the exams. To assist readers with those topics, the authors create new materials clarifying and expanding on those troublesome exam topics. As mentioned in the Introduction, the additional content about the exam is contained in a PDF on this book's companion website, at

<http://www.ciscopress.com/title/9780789760494>.

This appendix is intended to provide you with updated information if Amazon makes minor modifications to the exam upon which this book is based. When Amazon releases an entirely new exam, the changes are usually too extensive to provide in a simple update appendix. In those cases, you might need to consult the new edition of the book for the updated content. This appendix attempts to fill the void that occurs with any print book. In particular, this appendix does the following:

- ■ Mentions technical items that might not have been mentioned elsewhere in the book
 - ■ Covers new topics if Amazon adds new content to the exam over time
 - ■ Provides a way to get up-to-the-minute current information about content for the exam
-

ALWAYS GET THE LATEST AT THE BOOK'S PRODUCT PAGE

You are reading the version of this appendix that was available when your book was printed. However, given that the main purpose of this appendix is to be a living, changing document, it is important that you look for the latest version online at the book's companion website. To do so, follow these steps:

Step 1. Browse to

www.informit.com/title/9780789760494.

Step 2. Click the **Updates** tab.

Step 3. If there is a new Appendix B document on the page, download the latest Appendix B document.

Note

The downloaded document has a version number. Comparing the version of the print Appendix B (Version 1.0) with the latest online version of this appendix, you should do the following:

- ■ **Same version:** Ignore the PDF that you downloaded from the companion website.
- ■ **Website has a later version:** Ignore this Appendix B in your book and read only the latest version that you downloaded from the companion website.

TECHNICAL CONTENT

The current Version 1.0 of this appendix does not contain additional technical coverage.

Index

A

access

- ACLs (access control lists), [173](#), [175](#)
- authorization, [181–182](#)
- IAM (Identity and Access Management)
 - capabilities of, [169–171](#)
 - groups, [171](#)
 - policies, [172–173](#)
 - resource access authorization, [181–182](#)
 - roles, [172](#)
 - users, [171](#)

Pearson Test Prep practice test engine, [249–251](#)

ACLs (access control lists), [173](#), [175](#)

Active Directory, [33](#)

adaptive capacity, DynamoDB, [122](#)

addresses

 IP (Internet Protocol), [85](#), [193](#), [203–204](#)

 NAT (network address translation), [194](#), [205–206](#)

Advisor, Amazon Redshift, [134–135](#)

AKP (Asynchronous Key Prefetch), [116](#)

alarms, creating, [230–231](#)

alias records, [149–150](#)

allocated memory utilization, [232](#)

Amazon Aurora. *See* [Aurora](#)

Amazon CloudFront, [18](#), [99](#), [100](#), [147–149](#)

Amazon CloudWatch. *See* [CloudWatch](#)

Amazon Cognito, [77](#)

Amazon DynamoDB. *See* [DynamoDB](#)

Amazon Elastic Block Store. *See* [EBS \(Elastic Block Store\)](#)

Amazon Elastic Compute Cloud. *See* [EC2 \(Elastic Compute Cloud\) services](#)

Amazon Elastic Container Service. *See* [ECS \(Elastic Container Service\)](#)

Amazon Elastic File System. *See* [EFS \(Elastic File System\)](#)

Amazon Glacier. *See* [Glacier services](#)

Amazon Kinesis, [19](#)

Amazon Machine Images (AMIs), [47](#)

Amazon Redshift. *See* [Redshift](#)

Amazon Route 53. *See* [Route 53 service](#)

Amazon Simple Storage Service. *See* [S3 \(Simple Storage Service\)](#)

Amazon Virtual Private Cloud. *See* [VPC \(Virtual Private Cloud\)](#)

AMIs (Amazon Machine Images), [47](#)

APIs (application programming interfaces), support for, [7](#)

application security

IAM (Identity and Access Management)

- capabilities of, [169–171](#)
- groups, [171](#)
- policies, [172–173](#)
- resource access authorization, [181–182](#)
- roles, [172](#)

users, [171](#)
network ACLs (access control lists), [175](#)
security groups, [174](#)

application services
CloudFront, [18](#), [99](#), [100](#), [147–149](#)
HA (high availability), [88](#)
Kinesis, [19](#)
OpsWorks, [18](#)
SQS (Simple Queue Service), [19](#), [63–66](#)

architecture
HA (high availability), [81](#)
application services, [88](#)
Availability Zones, [85–87](#)
database services, [88–90](#)
EC2 instances, [85–88](#)
FT (fault tolerance), [84](#)
metrics, [84](#)
monitoring services, [93](#)
networking services, [91–92](#)
overview of, [84–85](#)
security services, [92–93](#)

multitier
advantages of, [74–75](#)
components of, [75–76](#)
three-tier architecture, [76–77](#)

single-tier
building with EC2, [72–74](#)
example of, [71–72](#)

SOA (service-oriented architecture), [74](#)
asynchronous decoupling
 overview of, [62–63](#)
SQS (Simple Queue Service), [63–66](#)
Asynchronous Key Prefetch (AKP), [116](#)
Aurora
 AKP (Asynchronous Key Prefetch), [116](#)
 Aurora Serverless, [114](#)
 DB connections, showing, [115](#)
 hash joins, [117](#)
 multithreaded replication, [116](#)
 overview of, [20, 114–115](#)
 read scaling capabilities, [117](#)
 T2 instances, [115–116](#)
 TCP keepalive parameters, [117–118](#)
AuroraReplicaLag, [116](#)
authoritative name servers, [150](#)
authorization, [181–182](#)
Auto Scaling, [16–17, 88, 160–163](#)
Availability Zones. *See* [HA \(high availability\) architecture](#)
AWS Batch, [14](#)
AWS Cloud Compliance, [239](#)
AWS CloudFormation, [17](#)
AWS CloudTrail. *See* [CloudTrail](#)
AWS Config, [240](#)
AWS Cost Explorer, [228–230](#)
AWS Database Migration Service, [23](#)
AWS Direct Connect, [25, 91–92, 185](#)

AWS Directory Service, [33](#)
AWS Elastic Beanstalk, [14–15](#)
AWS Elastic MapReduce, [184](#)
AWS Identity and Access Management. *See* [IAM \(Identity and Access Management\)](#)
AWS Key Management Service, [33](#), [99](#), [182–183](#)
AWS Lambda services, [230–231](#), [240](#)
AWS Management Console
 Lifecycle Management, [216–218](#)
 SQS (Simple Queue Service) configuration, [63–66](#)
AWS network infrastructure. *See* [network infrastructure](#)
AWS OpsWorks, [18](#)
AWS Security Token Service (AWS STS), [170](#)
AWS Serverless Application Repository, [13](#)
AWS Simple Monthly calculator, [214–216](#)
AWS Snowball, [30](#)
AWS Storage Gateway, [31](#)
AWS Systems Manager
 Patch Manager, [175–176](#)
 Run Command, [240](#)
AWS Trusted Advisor, [33–34](#), [239](#)
AWS Web Application Firewall, [32](#)
AWS Well-Architected Framework. *See* [Well-Architected Framework](#)
AWS X-Ray, [240](#)

B

background write process (ElastiCache), [129–130](#)

balancing load. *See* [ELB \(Elastic Load Balancing\)](#)

Batch, [14](#)

Billed Duration value, [232](#)

billing. *See also* [cost optimization](#)

alarms, creating, [230–231](#)

billed duration utilization, [232](#)

metrics, [230](#)

BitBucket, [242](#)

black boxes, [61](#)

BRPOPLPUSH command, [132](#)

buckets (S3)

creating, [44–46](#), [147](#), [216](#)

deleting, [47](#), [149](#), [218](#)

bulk retrieval (Glacier), [108](#), [219](#)

burst capacity (DynamoDB), [121](#)

BurstCreditBalance metric, [106](#)

Bursting Throughput mode (EFS), [105–106](#), [219](#)

C

caching

CloudFront, [18](#), [99](#), [100](#), [147–149](#)

DAX (DynamoDB Accelerator), [145–146](#)

ElastiCache

background write process, [129–130](#)

lazy loading, [127–128](#)

online cluster resizing, [131–132](#)

overview of, [22](#), [77](#), [126–127](#), [142–145](#)

reserved memory, [130–131](#)

TTL (time to live), [129](#)
write through, [128–129](#)

Greengrass, [149–150](#)

Route 53 service, [26](#), [150–153](#), [160](#)

calendars, Simple Monthly, [214–216](#)

CapEx, [6](#)

CASE expression, [134](#)

CI/CD (Continuous Integration/Continuous Deployment) pipeline, [239](#)

classes, S3 (Simple Storage Service), [42–44](#)

classic three-tier architecture, [76–77](#)

ClassicLink, [194](#), [206](#)

clearing data, [184](#)

Cloud Compliance, [239](#)

CloudFormation, [17](#)

CloudFront, [18](#), [99](#), [100](#), [147–149](#)

CloudTrail

- API activity tracking, [242](#)
- overview of, [34](#)
- storing encryption keys in, [182–183](#)

CloudWatch

- baselines, [241](#)
- billing alarms, [230–231](#)
- BurstCreditBalance metric, [106](#)
- capabilities of, [160](#)
- data aggregation in, [93](#)
- HA (high availability), [93](#)
- operational health monitoring, [241](#)

overview of, [34](#)

clusters

- DAX (DynamoDB Accelerator), [146](#)
- Redis ElastiCache, [131–132](#), [142–145](#)

Cognito service, [77](#)

Cold HDD, [49–51](#), [102](#)

compute services

- Auto Scaling
 - capabilities of, [16–17](#), [88](#), [160–161](#)
 - target tracking scaling policies, [162–163](#)
- Batch, [14](#)
- CloudFormation, [17](#)
- EC2 (Elastic Compute Cloud) services
 - Auto Scaling, [88](#)
 - cost optimization, [227–231](#)
 - HA (high availability) architecture, [85–88](#)
 - instances, [72–74](#), [85–88](#)
 - overview of, [8–10](#)
 - purpose of, [160](#)
 - resource access authorization, [181–182](#)
 - single-tier architecture, [72–74](#)
- ECR (Elastic Container Registry), [12–13](#)
- ECS (Elastic Container Service), [11–12](#), [160](#)
- EKS (Elastic Container Service for Kubernetes), [12–13](#)
- Elastic Beanstalk, [14–15](#)
- ELB (Elastic Load Balancing), [16](#), [62](#), [159–160](#)
- Fargate, [13](#)
- Lambda services

cost optimization, 230–231
metrics, 232
operational readiness and, 240
overview of, 10–11
Lightsail, 13–14
Serverless Application Repository, 13
Config, 240
configuration
CloudFront, 147–149
EFS (Elastic File System), 51–53
Greengrass, 149–150
Lifecycle Management, 216–218
Redis ElastiCache clusters, 131–132, 142–145
Route 53 service, 150–153
SQS (Simple Queue Service), 63–66
Configure Queue command, 64
Connect to Redis Server command, 145
container management
ECR (Elastic Container Registry), 12–13
ECS (Elastic Container Service), 11–12, 160
EKS (Elastic Container Service for Kubernetes), 12–13
Continuous Integration/Continuous Deployment (CI/CD)
pipeline, 239
contractual commitments, lack of, 6
convertible RIs (reserved instances), 227
cooldown periods, 163
COPY command, 133
Cost Explorer, 228–230

cost optimization

 EC2 (Elastic Compute Cloud) services

 billing alarms, 230–231

 considerations for, 227–228

 Cost Explorer, 228–230

 Lambda services, 231–232

 storage

 EBS (Elastic Block Store), 218

 EFS (Elastic File System), 218–219

 Glacier services, 219–221

 S3 (Simple Storage Service), 214–218

cp command, 100

CPUCreditBalance (Aurora), 115

Create Bucket command, 44, 147, 216

Create Distribution command, 148

Create File System command, 51

Create Hosted Zone command, 152

Create New Queue command, 64

Create Queue command, 66

Create Record Set command, 152

Create Vault command, 54, 220

Create Volume command, 49

D

Data Requests charge (S3), 216

Data Retrieval charge (S3), 216

data security

 data at rest, 183–184

data in transit, [185](#)
decommissioning process, [184](#)
encryption keys, storing in cloud, [182–183](#)
resource access authorization, [181–182](#)
SSL (Secure Sockets Layer), [185](#)
VPC (Virtual Private Cloud), [185](#)
Data Security Standard (DSS), [170](#)
data stores
 IAM-enabled options, [77](#)
 VPC-hosted options, [77](#)
data tier, [77](#)
Database Migration Service, [23](#)
database services
 Aurora
 AKP (Asynchronous Key Prefetch), [116](#)
 Aurora Serverless, [114](#)
 DB connections, showing, [115](#)
 hash joins, [117](#)
 multithreaded replication, [116](#)
 overview of, [20](#), [114–115](#)
 read scaling capabilities, [117](#)
 T2 instances, [115–116](#)
 TCP keepalive parameters, [117–118](#)
 Database Migration Service, [23](#)
 DynamoDB
 adaptive capacity, [122](#)
 burst capacity, [121](#)
 overview of, [21–22](#), [77](#)

query and design patterns, [119–121](#)

query operations, [124–126](#)

scan operations, [124–126](#)

secondary indexes, [122–124](#)

ElastiCache

background write process, [129–130](#)

configuration, [142–145](#)

lazy loading, [127–128](#)

online cluster resizing, [131–132](#)

overview of, [22, 77, 126–127](#)

reserved memory, [130–131](#)

TTL (time to live), [129](#)

write through, [128–129](#)

HA (high availability), [88–90](#)

RDS (Relational Database Services)

best practices, [118–119](#)

HA (high availability) architecture, [88–90](#)

overview of, [20–21, 77](#)

Redshift

Amazon Redshift Advisor, [134–135](#)

best practices, [132–134](#)

overview of, [22, 77](#)

DAX (DynamoDB Accelerator), [145–146](#)

decommissioning data, [184](#)

decoupling

advantages of, [62](#)

asynchronous, [62–63](#)

definition of, [61](#)

SQS (Simple Queue Service), [63–66](#)
 synchronous, [62](#)
delegation sets, [151](#)
Delete Bucket command, [47, 149, 218](#)
Delete File System command, [53](#)
Delete Hosted Zone command, [153](#)
Delete Vault command, [55, 221](#)
Delete Volume command, [49](#)
deleting
 EBS (Elastic Block Store) volumes, [49](#)
 EFS (Elastic File System) file systems, [53](#)
 Glacier vaults, [55, 221](#)
 hosted zones, [153](#)
 queues, [66](#)
 S3 (Simple Storage Service) buckets, [47, 149, 218](#)
On-Demand requests (Glacier), [108](#)
DescribeDBInstances action, [89](#)
describe-db-instances command, [89](#)
design patterns, DynamoDB, [119–121](#)
design principles, [237–238, 239–240](#)
destroying data, [184](#)
DHCP (Dynamic Host Control Protocol) option sets, [193, 202](#)
Direct Connect, [25, 91–92, 185](#)
Directory Service, [33](#)
distribution, CloudFront, [147–149](#)
DNS (Domain Name System)
 overview of, [193, 202–203](#)
 private, [151](#)

records, 151, 152–153
resolver, 150–151
DSS (Data Security Standard), 170
Duration value, 232
Dynamic Host Control Protocol. *See* [DHCP \(Dynamic Host Control Protocol\) option sets](#)
DynamoDB
adaptive capacity, 122
burst capacity, 121
DAX (DynamoDB Accelerator), 145–146
overview of, 21–22, 77
query and design patterns, 119–121
query operations, 124–126
scan operations, 124–126
secondary indexes, 122–124
DynamoDB Accelerator (DAX), 145–146

E

EBS (Elastic Block Store)
cost optimization, 218
EBS-optimized instances, 101–103
instance stores compared to, 47
overview of, 28–29
performance of, 101–103
resiliency, 47
volume creation, 48–49
volume deletion, 49
volume types, 49–51

volumes

 creating, [48–49](#)

 deleting, [49](#)

 types of, [49–51](#)

EC2 (Elastic Compute Cloud) services

 Auto Scaling, [88](#)

 cost optimization

 billing alarms, [230–231](#)

 considerations for, [227–228](#)

 Cost Explorer, [228–230](#)

 HA (high availability) architecture, [85–88](#)

 instances

 launching, [72–73](#)

 provisioning in different Availability Zones, [85–87](#)

 terminating, [74](#), [88](#)

 viewing, [73](#)

 overview of, [8–10](#)

 purpose of, [160](#)

 resource access authorization, [181–182](#)

 single-tier architecture, building, [72–74](#)

ECR (Elastic Container Registry), [12–13](#)

ECS (Elastic Container Service), [11–12](#), [160](#)

edge locations, [147](#)

EFS (Elastic File System)

 basic configuration, [51–53](#)

 cost optimization, [218–219](#)

File Sync, [219](#)

file systems

creating, [51–53](#)
deleting, [53](#)
overview of, [29, 51](#)
performance of, [103–107](#)
 Bursting Throughput mode, [105–106](#)
 General Purpose mode, [105](#)
 high-level characteristics, [103–104](#)
 performance tips, [107](#)
 Provisioned Throughput mode, [105–106](#)
resiliency, [51–53](#)
egress-only Internet gateways, [193, 201–202](#)
EKS (Elastic Container Service for Kubernetes), [12–13](#)
Elastic Beanstalk, [14–15](#)
Elastic Block Store. *See* [EBS \(Elastic Block Store\)](#)
Elastic Compute Cloud. *See* [EC2 \(Elastic Compute Cloud\)](#)
services
Elastic Container Registry (ECR), [12–13](#)
Elastic Container Service (ECS), [11–12, 160](#)
Elastic Container Service for Kubernetes (EKS), [12–13](#)
Elastic File System. *See* [EFS \(Elastic File System\)](#)
elastic IP addresses, [85](#)
Elastic Load Balancing (ELB), [62, 159–160](#)
Elastic MapReduce, [184](#)
ElastiCache
 background write process, [129–130](#)
 lazy loading, [127–128](#)
 online cluster resizing, [131–132](#)
 overview of, [22, 77, 126–127, 142–145](#)

reserved memory, 130–131

TTL (time to live), 129

write through, 128–129

elasticity

Auto Scaling

capabilities of, 16–17, 88, 160–161

cooldown periods, 163

target tracking scaling policies, 162–163

definition of, 6–7, 157

EBS (Elastic Block Store)

cost optimization, 218

EBS-optimized instances, 101–103

instance stores compared to, 47

overview of, 28–29

performance of, 101–103

resiliency, 47–51

volumes, 48–51

EC2 (Elastic Compute Cloud) services

Auto Scaling, 88

cost optimization, 227–231

HA (high availability) architecture, 85–88

instances, 72–74, 85–88

overview of, 8–10

purpose of, 160

resource access authorization, 181–182

single-tier architecture, 72–74

single-tier architecture, building, 72–74

ECR (Elastic Container Registry), 12–13

ECS (Elastic Container Service), [11–12](#), [160](#)

EFS (Elastic File System)

basic configuration, [51–53](#)

cost optimization, [218–219](#)

File Sync, [219](#)

file system creation/deletion, [51–53](#)

overview of, [29](#), [51](#)

performance of, [103–107](#)

resiliency, [51–53](#)

EKS (Elastic Container Service for Kubernetes), [12–13](#)

Elastic Beanstalk, [14–15](#)

elastic IP addresses, [85](#), [193](#), [203–204](#)

Elastic MapReduce, [184](#)

ElastiCache

background write process, [129–130](#)

configuration, [142–145](#)

lazy loading, [127–128](#)

online cluster resizing, [131–132](#)

overview of, [22](#), [77](#), [126–127](#)

reserved memory, [130–131](#)

TTL (time to live), [129](#)

write through, [128–129](#)

Elasticsearch, [77](#), [241](#)

ELB (Elastic Load Balancing), [16](#), [62](#), [159–160](#)

TCP window scaling, [99](#)

Elasticsearch, [77](#), [241](#)

ELB (Elastic Load Balancing), [16](#), [62](#), [159–160](#)

ENCODE parameter, [135](#)

EFS (Elastic File System), [107](#)
keys
 KMI (key management infrastructure), [183–184](#)
 KMS (Key Management Service), [33](#), [99](#), [182–183](#)
 storing in cloud, [182–183](#)
endpoints, VPC (Virtual Private Cloud), [194](#), [204–205](#)
Enterprise Support, [239](#)
ephemeral stores, [47](#)
events, responding to, [241](#)
Everything as a Service (XaaS), [7](#)
Evolve area of operational health, [242](#)
exam preparation
 chapter-ending review tools, [253](#)
 exam modes, [251](#)
 exam objectives and structure, [245–247](#)
 exam updates, [252](#), [273–274](#)
 memory tables
 how to use, [252–253](#)
Pearson Test Prep practice test engine
 offline access, [250–251](#)
 online access, [249–250](#)
 Premium Edition, [252](#)
 updates, [252](#)
suggested study plan, [253](#)
tips and suggestions, [248–249](#)
--exclude filter, [100](#)
expedited retrieval (Glacier), [108](#), [219](#)
experience, learning from, [242](#)

EXPLAIN statement, [117](#)

F

failovers, [90](#), [151](#)

Fargate, [13](#)

fault tolerance (FT), [84](#)

File Sync (EFS), [219](#)

file systems

 EFS (Elastic File System)

 basic configuration, [51–53](#)

 cost optimization, [218–219](#)

 File Sync, [219](#)

 file system creation/deletion, [51–53](#)

 overview of, [29](#), [51](#)

 performance of, [103–107](#)

 resiliency, [51–53](#)

 NFS (Network File System), [38](#), [51](#)

firewalls, WAF (Web Application Firewall), [32](#)

Flash Card mode, [251](#)

flexibility. *See* [elasticity](#)

FLUSHALL command, [132](#)

FLUSHDB command, [132](#)

FreeableMemory (Aurora), [116](#)

FT (fault tolerance), [84](#)

G

gateways

 definition of, [193](#)

 egress-only, [193](#), [201–202](#)

gateway endpoints, 205
overview of, 201
Storage Gateway service, 31

General Purpose mode (EFS), 105

General Purpose SSD, 49–51, 103

geolocation routing policy, 151

GET requests, 99

GitHub, 242

Glacier services

- cost optimization, 219–221
- overview of, 30
- performance of, 108
- resiliency, 53–55
 - retrieval policies, 55
 - vault creation, 54–55
 - vault deletion, 55
- retrieval policies, 55

vaults

- creating and mounting, 54–55
- deleting, 55

global infrastructure, 7, 23–24

global secondary indexes, 121, 122–124

Grant Public Read Access to This Object(s) command, 148

Greengrass, 149–150

GROUP BY clause, 134

groups

- IAM (Identity and Access Management), 171
- security, 174

“Guidelines for Media Sanitization”, [184](#)

H

HA (high availability) architecture

application services, [88](#)

Availability Zones, [85–87](#)

database services, [88–90](#)

EC2 instances, [85–88](#)

FT (fault tolerance), [84](#)

metrics, [84](#)

monitoring services, [93](#)

networking services, [91–92](#)

overview of, [84–85](#)

S3 (Simple Storage Service), [42](#)

security services, [92–93](#)

Hadoop, [184](#)

hash joins, [117](#)

health, operational, [241](#)

high availability. *See HA (high availability) architecture*

hosted zones, [151](#)

creating, [152–153](#)

deleting, [153](#)

HTTPS, [185](#)

I

IaaS (infrastructure as a service), [7](#)

IAM (Identity and Access Management)

capabilities of, [169–171](#)

data store options, [77](#)

groups, [171](#)
HA (high availability) architecture, [92–93](#)
overview of, [31–32](#)
policies, [172–173](#)
resource access authorization, [181–182](#)
roles, [172](#)
users, [171](#)

identities (IAM)
groups, [171](#)
policies, [172–173](#)
roles, [172](#)
users, [171](#)

Identity and Access Management. *See* [IAM \(Identity and Access Management\)](#)

identity-based policies, [173](#)

IIS (Internet Information Server), [71](#)

images, AMIs (Amazon Machine Images), [47](#)

indexes, [122–124](#)

infrastructure as a service (IaaS), [7](#)

initialization, EBS (Elastic Block Store), [102](#)

innodb_read_only global variable, [115](#)

input/output operations per second (IOPS), [101](#)

instance stores, [47](#)

instances
EBS-optimized, [101–103](#)

EC2 (Elastic Compute Cloud) services
HA (high availability) architecture, [85–88](#)
launching, [72–73](#)

provisioning in different Availability Zones, [85–87](#)
terminating, [74](#), [88](#)
viewing, [73](#)

RIs (reserved instances), [227–228](#)
T2, [115–116](#)

interdependencies, reducing. *See decoupling*
interface endpoints, [204–205](#)
interfaces, [193](#), [198–199](#)
Internet gateways. *See gateways*
Internet Information Server (IIS), [71](#)
Internet of Things (IoT), [149–150](#)
I/O requests, [102](#)
IOPS (input/output operations per second), [101](#)
IoT (Internet of Things), [149–150](#)
IP addresses, [85](#), [193](#), [203–204](#)

J-K

Key Management Service. *See KMS (Key Management Service)*

keys
AKP (Asynchronous Key Prefetch), [116](#)
KMI (key management infrastructure), [183–184](#)
KMS (Key Management Service), [33](#), [99](#), [182–183](#)
storing in cloud, [182–183](#)

KEYS command, [132](#)

Kinesis, [19](#)

KMI (key management infrastructure), [183–184](#)

KMS (Key Management Service), [33](#), [99](#), [182–183](#)

Kubernetes, EKS (Elastic Container Service for Kubernetes),
[12–13](#)

L

Lambda services

cost optimization, 230–231

metrics, 232

operational readiness and, 240

overview of, 10–11

latency. *See also* performant databases; performant storage

EBS (Elastic Block Store), 102–103

EFS (Elastic File System), 104–107

S3 (Simple Storage Service), 99–100

latency routing policy, 151

Launch Cost Explorer link, 228

Launch Instance command, 72

lazy loading, 127–128

learning

from experience, 242

sharing, 242

Lifecycle Management

Glacier services, 220–221

S3 (Simple Storage Service), 216–218

Lightsail, 13–14

LIKE operators, 134

lists, access control, 175

load balancing, ELB (Elastic Load Balancing), 16, 62, 159–160

loading, lazy, [127–128](#)
local secondary indexes, [122–124](#)
Loggly, [241](#)
logic tier, [77](#)
loose coupling, [62](#), [63](#)
Lua, [132](#)

M

Management Console
Lifecycle Management, [216–218](#)
SQS (Simple Queue Service) configuration, [63–66](#)
management services

CloudTrail
API activity tracking, [242](#)
overview of, [34](#)
storing encryption keys in, [182–183](#)

CloudWatch
baselines, [241](#)
billing alarms, [230–231](#)
billing alarms, creating, [230–231](#)
BurstCreditBalance metric, [106](#)
capabilities of, [160](#)
data aggregation in, [93](#)
HA (high availability), [93](#)
operational health monitoring, [241](#)
overview of, [34](#)

Trusted Advisor, [33–34](#), [239](#)
MariaDB, multi-AZ deployments for, [88](#)

master keys
 storing in cloud, 182–183

Max Memory Used value, 232

media sanitization, 184

memcached, 142

memory allocation, 232. *See also* [caching](#)

Memory Size value, 232

memory tables
 how to use, 252–253

metrics
 billing, 230
 HA (high availability), 84
 Lambda, 232
 target tracking scaling policies, 162–163

MFA (multifactor authentication), 169–170

Microsoft Active Directory, 33

migration, Database Migration Service, 23

monitoring services, high availability, 93

mounting Glacier vaults, 54–55

Multi-AZ deployments, 88–90

multifactor authentication (MFA), 169–170

multithreaded replication, 116

multitier architecture
 advantages of, 74–75
 components of, 75–76
 three-tier architecture, 76–77

multivalue answer routing policy, 151

mv command, 100

N

- name servers, authoritative, 150
- NAT (network address translation), 194, 205–206
- native Direct Connect, 91–92
- negotiations, reduction in, 6
- network ACLs (access control lists), 173, 175
- network address translation (NAT), 194, 205–206
- Network Data Transferred In charge (S3), 216
- Network Data Transferred Out charge (S3), 216
- Network File System (NFS), 38, 51, 107
- network infrastructure
 - ClassicLink, 194, 206
 - default components, checking, 194–198
 - DHCP (Dynamic Host Control Protocol) option sets, 193, 202
 - DNS (Domain Name System), 193, 202–203
 - elastic IP addresses, 193, 203–204
- gateways
 - definition of, 193
 - egress-only, 193, 201–202
 - gateway endpoints, 205
 - overview of, 201
 - Storage Gateway service, 31
- NAT (network address translation), 194, 205–206
- network interfaces, 193, 198–199
- overview of, 193–194
- route tables, 193, 199–200
- VPC (Virtual Private Cloud), 185

data store options, [77](#)
endpoints, [194](#), [204–205](#)
overview of, [24](#)
peering, [194](#), [206](#)
network interfaces, [193](#), [198–199](#)
networking services
AWS global infrastructure, [23–24](#)
Direct Connect, [25](#), [91–92](#), [185](#)
HA (high availability), [91–92](#)
Route 53 service, [26](#), [150–153](#), [160](#)
VPC (Virtual Private Cloud), [185](#)
data store options, [77](#)
endpoints, [194](#), [204–205](#)
overview of, [24](#)
peering, [194](#), [206](#)
New Relic, [241](#)
NFS (Network File System), [38](#), [51](#), [107](#)
NIST “Guidelines for Media Sanitization”, [184](#)
nodes, DAX (DynamoDB Accelerator), [146](#)
n-tier architecture. *See* [multitier architecture](#)

O

object storage. *See* [storage services](#)
Object-Level Logging, [46](#)
one-tier architecture. *See* [single-tier architecture](#)
online cluster resizing (ElastiCache), [131–132](#)
online resources
book companion website, [273–274](#)

Pearson Test Prep practice test engine, [249–251](#)
Operate area of operational excellence, [241](#)
operational excellence
 design principles, [238, 239–240](#)
 Evolve area, [242](#)
 Operate area, [241](#)
 Prepare area, [239–240](#)
operational health, [241](#)
operational priorities, [239](#)
operational readiness, [240](#)
operators, LIKE, [134](#)
OpEx, [6](#)
OpsWorks, [18](#)
optimization, cost
 EC2 (Elastic Compute Cloud) services, [227–231](#)
 Lambda services, [231–232](#)
Oracle, multi-AZ deployments for, [88](#)

P

PaaS (platform as a service), [7](#)
Patch Manager, [175–176](#)
“pay as you go” model, [6](#)
Payment Card Industry (PCI), [170](#)
PCI (Payment Card Industry), [170](#)
Pearson Test Prep practice test engine
 offline access, [250–251](#)
 online access, [249–250](#)
Premium Edition, [252](#)

updates, 252

peering, VPC (Virtual Private Cloud), 194, 206

performance optimization. *See* [caching](#)

performant databases

Aurora

- AKP (Asynchronous Key Prefetch), 116
- Aurora Serverless, 114
- DB connections, showing, 115
- hash joins, 117
- multithreaded replication, 116
- overview of, 20, 114–115
- read scaling capabilities, 117
- T2 instances, 115–116
- TCP keepalive parameters, 117–118

DynamoDB

- adaptive capacity, 122
- burst capacity, 121
- query and design patterns, 119–121
- query operations, 124–126
- scan operations, 124–126
- secondary indexes, 122–124

ElastiCache

- background write process, 129–130
- lazy loading, 127–128
- online cluster resizing, 131–132
- overview of, 126–127
- reserved memory, 130–131
- TTL (time to live), 129

write through, [128–129](#)

RDS (Relational Database Services), [118–119](#)

Redshift

Amazon Redshift Advisor, [132–135](#)

best practices, [132–134](#)

overview of, [22, 77](#)

performant storage

EBS (Elastic Block Store)

cost optimization, [218](#)

EBS-optimized instances, [101–103](#)

instance stores compared to, [47](#)

overview of, [28–29](#)

performance of, [101–103](#)

resiliency, [47–51](#)

volumes, [48–51](#)

EFS (Elastic File System)

basic configuration, [51–53](#)

Bursting Throughput mode, [105–106](#)

cost optimization, [218–219](#)

File Sync, [219](#)

file system creation/deletion, [51–53](#)

General Purpose mode, [105](#)

high-level characteristics, [103–104](#)

overview of, [29, 51](#)

performance of, [103–107](#)

performance tips, [107](#)

Provisioned Throughput mode, [105–106](#)

resiliency, [51–53](#)

Glacier services

cost optimization, 219–221

overview of, 30

performance of, 108

resiliency, 53–55

retrieval policies, 55

vault creation, 54–55

vault deletion, 55

S3 (Simple Storage Service)

advantages of, 42

bucket creation, 44–46, 147, 216

bucket deletion, 47, 149, 218

cost optimization, 214–218

overview of, 26–28, 77

performance of, 99–101

resiliency, 42–47

permissions, 46, 170

platform as a service (PaaS), 7

policies

IAM (Identity and Access Management), 172–173

routing, 151

scaling, 162–163

PostgreSQL, multi-AZ deployments for, 88

Practice Exam mode, 251

practice test engine (Pearson Test Prep)

offline access, 250–251

online access, 249–250

Premium Edition, 252

updates, 252

Premium Edition (Pearson Test Prep), 252

preparation, operational excellence, 239–240

Prepare area of operational excellence, 239–240

presentation tier, 76

pre-warming, 102

pricing

- EBS (Elastic Block Store), 218
- EC2 (Elastic Compute Cloud) services
 - billing alarms, 230–231
 - considerations for, 227–228
 - Cost Explorer, 228–230
- EFS (Elastic File System), 218–219
- Glacier services, 219–221
- Lambda services, 231–232
- S3 (Simple Storage Service), 214–218

priorities, operational, 239

private DNS, 151

Provisioned Capacity (Glacier), 108

Provisioned IOPS SSD, 49–51, 103

Provisioned Throughput mode (EFS), 105–106

purging data, 184

Q

queries, DynamoDB, 124–126

query patterns, DynamoDB, 119–121

queues

- creating, 63–65

deleting, [66](#)

SQS (Simple Queue Service)

 configuration, [63–66](#)

 overview of, [19](#)

R

RDS (Relational Database Services)

 best practices, [118–119](#)

 HA (high availability) architecture, [88–90](#)

 overview of, [20–21, 77](#)

read scaling, [117](#)

readiness, operational, [240](#)

records

 alias, [149–150](#)

 DNS (Domain Name System), [151, 152–153](#)

Recovery Point Objective (RPO), [84](#)

Recovery Time Objective (RTO), [84](#)

Redis, ElastiCache for

 background write process, [129–130](#)

 lazy loading, [127–128](#)

 online cluster resizing, [131–132](#)

 overview of, [22, 77, 126–127, 142–145](#)

 reserved memory, [130–131](#)

 TTL (time to live), [129](#)

 write through, [128–129](#)

Redshift

 Amazon Redshift Advisor, [132–135](#)

 best practices, [132–134](#)

overview of, [22](#), [77](#)

registry, ECR (Elastic Container Registry), [12–13](#)

Relational Database Services. *See* [RDS \(Relational Database Services\)](#)

replication, multithreaded, [116](#)

requests

- EBS (Elastic Block Store), [102](#)
- EFS (Elastic File System), [107](#)
- Glacier services, [108](#)
- S3 (Simple Storage Service), [99](#)

reserved instances (RIs), [227–228](#)

reserved memory (ElastiCache), [130–131](#)

resharding (ElastiCache), [131–132](#)

resiliency, [39](#)

EBS (Elastic Block Store), [47](#)

- instance stores compared to, [47](#)
- volume creation, [48–49](#)
- volume deletion, [49](#)
- volume types, [49–51](#)

EFS (Elastic File System)

- basic configuration, [51–53](#)
- file systems, [51–53](#)
- overview of, [51](#)

Glacier services, [53–55](#)

- retrieval policies, [55](#)
- vault creation, [54–55](#)
- vault deletion, [55](#)

S3 (Simple Storage Service)

advantages of, [42](#)
bucket creation, [44–46](#)
bucket deletion, [47](#)
capabilities of, [42](#)
classes, [42–44](#)
storage classes, [42–44](#)
resolver (DNS), [150–151](#)
resource access authorization, [181–182](#)
resource-based policies, [173](#)
responses to events, [241](#)
rest, protecting data at, [183–184](#)
retrieval policies, Glacier, [55](#), [108](#)
reusable delegation sets, [151](#)
RIs (reserved instances), [227–228](#)
roles, IAM (Identity and Access Management), [172](#)
root users, [169](#)
Route 53 service, [26](#), [150–153](#), [160](#)
route tables, [193](#), [199–200](#)
routing policy, [151](#)
RPO (Recovery Point Objective), [84](#)
RTO (Recovery Time Objective), [84](#)

S

S3 (Simple Storage Service). *See also* [Glacier services](#)

advantages of, [42](#)
buckets
 creating, [44–46](#), [147](#), [216](#)
 deleting, [47](#), [149](#), [218](#)

cost optimization, 214
estimated costs, 214–216
Lifecycle Management, 216–218
HA (high availability) architecture, 42
overview of, 26–28, 77
performance of, 99–101
resiliency
 bucket creation, 44–46
 bucket deletion, 47
 capabilities of, 42
 storage classes, 42–44
S3 Glacier class, 43–44
S3 One Zone-Infrequent Access class, 43–44
S3 Standard class, 42–44
S3 Standard-Infrequent Access class, 43–44
SaaS (software as a service), 7
sanitization, media, 184
scalability, 157
 Auto Scaling
 capabilities of, 16–17, 88, 160–161
 cooldown periods, 163
 target tracking scaling policies, 162–163
ELB (Elastic Load Balancing), 159–160
S3 (Simple Storage Service), 42
TCP windows, 99
scale-in cooldown period, 163
scale-out cooldown period, 163
SCAN command, 132

scans, DynamoDB, 124–126

scheduled RIs (reserved instances), 227

SCPs (Service Control Policies), 173

secondary indexes, 122–124

Secure Sockets Layer (SSL), 185

security groups, 174

security services. *See also* encryption

- ACLs (access control lists), 173
- data security
 - data at rest, 183–184
 - data in transit, 185
 - decommissioning process, 184
 - encryption keys, 182–183
 - resource access authorization, 181–182
 - SSL (Secure Sockets Layer), 185
 - VPC (Virtual Private Cloud), 185

Directory Services, 33

IAM (Identity and Access Management)

- capabilities of, 169–171
- data store options, 77
- groups, 171
- HA (high availability) architecture, 92–93
- overview of, 31–32
- policies, 172–173
- resource access authorization, 181–182
- roles, 172
- users, 171

KMS (Key Management Service), 33, 99, 182–183

network ACLs (access control lists), [175](#)
security groups, [174](#)
Systems Manager Patch Manager, [175–176](#)
WAF (Web Application Firewall), [32](#)
Security Token Service (STS), [170–171](#)
selective acknowledgement (TCP), [99](#)
Send a Message command, [65](#)
server access logging, [46](#)
Serverless Application Repository, [13](#)
Service Control Policies (SCPs), [173](#)
Service Health dashboard, [241](#)
service-oriented architecture (SOA), [74](#)
sharing learning, [242](#)
Simple Monthly calculator, [214–216](#)
Simple Queue Service (SQS), [19](#), [63–66](#)
simple routing policy, [151](#)
Simple Storage Service. *See* S3 (Simple Storage Service)
single-tier architecture
 building with EC2, [72–74](#)
 example of, [71–72](#)
SMEMBERS command, [132](#)
Snowball, [30](#)
SOA (service-oriented architecture), [74](#)
software as a service (SaaS), [7](#)
SourceForge, [242](#)
Splunk, [241](#)
SQL Server Mirroring, [88](#)
SQS (Simple Queue Service), [19](#), [63–66](#)

SSCAN command, 132

SSL (Secure Sockets Layer), 185

standard retrieval (Glacier), 108, 219

standard RIs (reserved instances), 227

stdin command, 100–101

stdout command, 100–101

Storage Gateway, 31

storage services

EBS (Elastic Block Store)

- cost optimization, 218
- EBS-optimized instances, 101–103
- instance stores compared to, 47
- overview of, 28–29
- performance of, 101–103
- resiliency, 47–51
- volume creation, 48–49
- volume deletion, 49
- volume types, 49–51
- volumes, 48–51

EFS (Elastic File System)

- basic configuration, 51–53
- Bursting Throughput mode, 105–106
- cost optimization, 218–219
- File Sync, 219
- file system creation/deletion, 51–53
- file systems, 51–53
- General Purpose mode, 105
- high-level characteristics, 103–104

overview of, [29](#), [51](#)
performance of, [103–107](#)
performance tips, [107](#)
Provisioned Throughput mode, [105–106](#)
resiliency, [51–53](#)

Glacier services

cost optimization, [219–221](#)
overview of, [30](#)
performance of, [108](#)
resiliency, [53–55](#)
retrieval policies, [55](#)
vault creation, [54–55](#)
vault deletion, [55](#)

S3 (Simple Storage Service)

advantages of, [42](#)
bucket creation, [44–46](#), [147](#), [216](#)
bucket deletion, [47](#), [149](#), [218](#)
classes, [42–44](#)
cost optimization, [214–218](#)
overview of, [26–28](#), [77](#)
performance of, [99–101](#)
resiliency, [42–47](#)

Snowball, [30](#)

Storage Gateway, [31](#)

Storage Used charged (S3), [215](#)

stores

ephemeral, [47](#)
instance, [47](#)

STS (Security Token Services), [171](#)
Study mode, [251](#)
study plan for exam, [253](#)
SumoLogic, [241](#)
sync command, [100](#)
synchronous decoupling, [62](#)
Systems Manager
 Patch Manager, [175–176](#)
 Run Command, [240](#)
Systems Manager Automation, [240](#)

T

T2 instances, [115–116](#)
tables, route, [193, 199–200](#)
TAMs (Technical Account Managers), [239](#)
target tracking scaling policies, [162–163](#)
TCP (Transmission Control Protocol)
 keepalive parameters, [117–118](#)
 selective acknowledgement, [99](#)
 window scaling, [99](#)
Technical Account Managers (TAMs), [239](#)
terminating EC2 instances, [74, 88](#)
three-tier architecture, [76–77](#)
throughput, EFS (Elastic File System), [105–106](#)
Throughput Optimized HDD, [49–51, 102](#)
time to live (TTL), [129, 151](#)
transit, data protection during, [185](#)
Transmission Control Protocol. *See* TCP (Transmission

Control Protocol)

Trusted Advisor, [33–34](#), [239](#)
TTL (time to live), [129](#), [151](#)

U

updates, exam, [252](#), [273–274](#)
updates, software, [175–176](#)
users
IAM (Identity and Access Management), [171](#)
root, [169](#)

V

vaults, Glacier
creating and mounting, [54–55](#)
deleting, [55](#)
retrieval rate for, [220–221](#)
View Instances command, [73](#)
Virtual Private Cloud. *See* [VPC \(Virtual Private Cloud\)](#)
virtual private networks (VPNs), [91–92](#), [185](#)
volume queue length, [103](#)
volumes (EBS)
creating, [48–49](#)
deleting, [49](#)
types of, [49–51](#)
VPC (Virtual Private Cloud)
data security, [185](#)
data store options, [77](#)
endpoints, [194](#), [204–205](#)
overview of, [24](#)

peering, 194, 206
VPNs (virtual private networks), 91–92, 185

W

WAF (Web Application Firewall), 32
weighted round robin, 151
Well-Architected Framework
 operational excellence
 design principles, 237–238, 239–240
 Evolve area, 242
 Operate area, 241
 Prepare area, 239–240
 overview of, 237–238
WHERE clause, 134
write through, 128–129
WSCALE factor, 99

X-Y-Z

XaaS (Everything as a Service), 7
X-Ray, 240
zone apex, 150
zones, hosted, 151
 creating, 152–153
 deleting, 153

Online Elements

Appendix C. Memory Tables

CHAPTER 2

Table 2-2 The S3 Classes

	S3 Standard	S3 Standard- IA	S3 One Zone-IA	Amazon Glacier
Durability	99.999999 99%			
Availability	99.99%			
Availability SLA	99.9%			
Availability Zones	3			
Storage Type	Object			

Lifecycle Transitions	Yes
----------------------------------	-----

CHAPTER 4

Table 4-2 Sample Components of an AWS Multitier Architecture

Requirement	Service
DNS Resolution	
Content Delivery Network	
Web Resources	
Web Servers	
Load Balancing	
Scalability	

Application Servers
Database Servers

CHAPTER 5

Table 5-2 S3 High Availability

S3 Standard	S3 Standard-IA	S3 One Zone-IA	Amazon Glacier
Availability	99.99%		
Availability Zones	Greater than or equal to three		

CHAPTER 6

Table 6-2 EFS vs. EBS

EFS	EBS Provisioned IOPS

Per-operation latency
Throughput scale
Availability and durability
Access

CHAPTER 7

Table 7-2 Advantages and Disadvantages of Lazy Loading

Advantages	Disadvantages

Table 7-3 Advantages and Disadvantages of Write Through

Advantages	Disadvantages

CHAPTER 11

You can use features of EC2 and Identity and Access Management (IAM) to

- _____

- _____

- _____

- _____

IAM enables you to do the following:

- _____

- 
 - 
 - 
 - 
 - 
 - 
 - 
 - 
 - 
 - 
 - 
 - 
 - 
 - 
 - 
 - 
 - 

You can perform the following management actions on your AWS KMS master keys:

- 
 - 
 - 
 - 
 - 
 - 
 - 
 - 
 - 
 - 
 - 
 - 
 - 
 - 
 - 
 - 
 - 
 - 
 - 
 - 
 - 
 - 
 - 

-
-

With AWS KMS, you can also perform the following cryptographic functions using master keys:

-
-
-

With AWS, there are three different models for how you and AWS provide the encryption of data at rest and the KMI.

These three models are

-
-
-

Appendix D. Memory Tables

Answer Key

CHAPTER 2

Table 2-2 The S3 Classes

	S3 Standard	S3 Standard- IA	S3 One Zone-IA	Amazon Glacier
Durability	99.999999 99%	99.999999 99%	99.999999 99%	99.9999 9999%
Availability	99.99%	99.9%	99.5%	N/A
Availability SLA	99.9%	99%	99%	N/A
Availability Zones	3	3	1	3

Storage Type	Object	Object	Object	Object
Lifecycle Transitions	Yes	Yes	Yes	Yes

CHAPTER 4

Table 4-2 Sample Components of an AWS Multitier Architecture

Requirement	Service
DNS Resolution	Route 53
Content Delivery Network	CloudFront
Web Resources	Simple Storage Service
Web Servers	Elastic Compute Cloud
Load Balancing	Elastic Load Balancing

Scalability	Auto Scaling
Application Servers	Elastic Compute Cloud
Database Servers	Relational Database Services

CHAPTER 5

Table 5-2 S3 High Availability

	S3 Standard	S3 Standard-IA	S3 One Zone-IA	Amazon Glacier
Availability	99.99%	99.9%	99.5%	NA
Availability Zones	Greater than or equal to three	Greater than or equal to three	One	Greater than or equal to three

CHAPTER 6

Table 6-2 EFS vs. EBS

EFS	EBS Provisioned IOPS
Per-operation latency	Low, consistent latency
Throughput scale	Up to 2 GB per second
Availability and durability	Redundant across multiple AZs
Access	Single instance

CHAPTER 7

Table 7-2 Advantages and Disadvantages of Lazy Loading

Advantages	Disadvantages
Only requested data is cached.	There is a cache miss penalty. Each cache miss results in three trips: the initial request for data from the cache; the query of the database for the data; and the

	writing of data to the cache, which can cause a noticeable delay in data getting to the application.
Because most data is never requested, lazy loading avoids filling up the cache with data that is not requested.	If data is written to the cache only when there is a cache miss, data in the cache can become <i>stale</i> because there are no updates to the cache when data is changed in the database. This issue is addressed by the write through and adding TTL strategies.
Node failures are not fatal.	

Table 7-3 Advantages and Disadvantages of Write Through

Advantages	Disadvantages
Data in the cache is never stale: Because the data in the cache is updated every time it is written to the database, the data in the cache is always current.	Missing data: In the case of spinning up a new node, whether due to a node failure or scaling out, there is missing data that continues to be missing until it is added or updated on the database. This situation can be minimized by implementing lazy loading in conjunction with write through.

Write penalty vs. Read penalty: Every write involves two trips: a write to the cache and a write to the database. This adds latency to the process. That said, end users are generally more tolerant of latency when updating data than when retrieving data. There is an inherent sense that updates are more work and thus take longer.

Cache churn: Because most data is never read, a lot of data in the cluster is never read. This is a waste of resources. By adding TTL, you can minimize wasted space.

CHAPTER 11

You can use features of EC2 and Identity and Access Management (IAM) to

- Allow other users, services, and applications to use your EC2 resources without sharing your security credentials
- Control how other users use resources in your AWS account
- Use security groups to control access to your EC2 instances
- Allow full use or limited use of your EC2 resources

IAM enables you to do the following:

- Create users and groups under your AWS account
- Assign unique security credentials to each user under your AWS account
- Control each user's permissions to perform tasks using AWS resources
- Allow the users in another AWS account to share your AWS resources
- Create roles for your AWS account and define the users or services that can assume them
- Use existing identities for your enterprise to grant permissions to perform tasks using AWS resources

You can perform the following management actions on your AWS KMS master keys:

- ■ Create, describe, and list master keys
- ■ Enable and disable master keys
- ■ Create and view grants and access control policies for your master keys
- ■ Enable and disable automatic rotation of the cryptographic material in a master key
- ■ Import cryptographic material into an AWS KMS master key
- ■ Tag your master keys for easier identification, categorizing, and tracking
- ■ Create, delete, list, and update aliases, which are friendly names associated with your master keys
- ■ Delete master keys to complete the key lifecycle

With AWS KMS, you can also perform the following cryptographic functions using master keys:

- ■ Encrypt, decrypt, and re-encrypt data
- ■ Generate data encryption keys that you can export from the service in plaintext or encrypted under a master key that does not leave the service
- ■ Generate random numbers suitable for cryptographic applications

With AWS, there are three different models for how you and AWS provide the encryption of data at rest and the KMI. These three models are

- ■ You control the encryption method and the entire KMI.
- ■ You control the encryption method, AWS provides the storage for KMI, and you control the management of the KMI.
- ■ AWS controls the encryption method and the entire KMI.

Appendix E. Study Planner

Practice Test	Reading	Task
---------------	---------	------

Element	Task	G o al D a te	Firs t Dat e Co mpl eted	Second Date Comple ted (Option al)	N o t e s
Introduction	Read Introduction				
1. The Fundamentals of AWS	Read Foundation Topics				
1. The Fundamentals of AWS	Review Key Topics				

1. The Fundamentals of AWS	Define Key Terms				
1. The Fundamentals of AWS	Complete Q&A section				
Practice Test	Take practice test in study mode using Exam Bank 1 questions for Chapter 1 in practice test software				
2. Designing Resilient Storage	Read Foundation Topics				
2. Designing Resilient Storage	Review Key Topics				
2. Designing Resilient Storage	Define Key Terms				
2. Designing	Complete Q&A section				

Resilient Storage					
Practice Test	Take practice test in study mode using Exam Bank 1 questions for Chapter 2 in practice test software				
3. Designing Decoupling Mechanisms	Read Foundation Topics				
3. Designing Decoupling Mechanisms	Review Key Topics				
3. Designing Decoupling Mechanisms	Define Key Terms				
3. Designing Decoupling Mechanisms	Complete Q&A section				
Practice Test	Take practice test in study mode using Exam Bank 1 questions for Chapter 3 in				

	practice test software				
4. Designing a Multitier Infrastructure	Read Foundation Topics				
4. Designing a Multitier Infrastructure	Review Key Topics				
4. Designing a Multitier Infrastructure	Define Key Terms				
4. Designing a Multitier Infrastructure	Complete Q&A section				
Practice Test	Take practice test in study mode using Exam Bank 1 questions for Chapter 4 in practice test software				
5. Designing High Availability Architectures	Read Foundation Topics				

5. Designing High Availability Architectures	Review Key Topics				
5. Designing High Availability Architectures	Define Key Terms				
5. Designing High Availability Architectures	Complete Q&A section				
Practice Test	Take practice test in study mode using Exam Bank 1 questions for Chapter 5 in practice test software				
6. Choosing Performant Storage	Read Foundation Topics				
6. Choosing Performant	Review Key Topics				

Storage					
6. Choosing Performant Storage	Define Key Terms				
6. Choosing Performant Storage	Complete Q&A section				
Practice Test	Take practice test in study mode using Exam Bank 1 questions for Chapter 6 in practice test software				
7. Choosing Performant Databases					
7. Choosing Performant Databases	Review Key Topics				
7. Choosing Performant Databases	Define Key Terms				

7. Choosing Performant Databases	Complete Q&A section				
Practice Test	Take practice test in study mode using Exam Bank 1 questions for Chapter 7 in practice test software				
8. Improving Performance with Caching	Read Foundation Topics				
8. Improving Performance with Caching	Review Key Topics				
8. Improving Performance with Caching	Define Key Terms				
8. Improving Performance with Caching	Complete Q&A section				

Practice Test	Take practice test in study mode using Exam Bank 1 questions for Chapter 8 in practice test software				
9. Designing for Elasticity	Read Foundation Topics				
9. Designing for Elasticity	Review Key Topics				
9. Designing for Elasticity	Define Key Terms				
9. Designing for Elasticity	Complete Q&A section				
Practice Test	Take practice test in study mode using Exam Bank 1 questions for Chapter 9 in practice test software				
10. Securing Application Tiers	Read Foundation Topics				

10. Securing Application Tiers	Review Key Topics				
10. Securing Application Tiers	Define Key Terms				
10. Securing Application Tiers	Complete Q&A section				
Practice Test	Take practice test in study mode using Exam Bank 1 questions for Chapter 10 in practice test software				
11. Securing Data	Read Foundation Topics				
11. Securing Data	Review Key Topics				
11. Securing Data	Define Key Terms				

11. Securing Data	Complete Q&A section				
Practice Test	Take practice test in study mode using Exam Bank 1 questions for Chapter 11 in practice test software				
12. Networking Infrastructure for a Single VPC Application	Read Foundation Topics				
12. Networking Infrastructure for a Single VPC Application	Review Key Topics				
12. Networking Infrastructure for a Single VPC Application	Define Key Terms				
12. Networking	Complete Q&A section				

Infrastructure for a Single VPC Application					
Practice Test	Take practice test in study mode using Exam Bank 1 questions for Chapter 12 in practice test software				
13. Cost-Optimized Storage	Read Foundation Topics				
13. Cost-Optimized Storage	Review Key Topics				
13. Cost-Optimized Storage	Define Key Terms				
13. Cost-Optimized Storage	Complete Q&A section				
Practice Test	Take practice test in study				

	mode using Exam Bank 1 questions for Chapter 13 in practice test software				
14. Cost-Optimized Compute	Read Foundation Topics				
14. Cost-Optimized Compute	Review Key Topics				
14. Cost-Optimized Compute	Define Key Terms				
14. Cost-Optimized Compute	Complete Q&A section				
Practice Test	Take practice test in study mode using Exam Bank 1 questions for Chapter 14 in practice test software				
15. Features for Operational	Read Foundation Topics				

Excellence					
15. Features for Operational Excellence	Review Key Topics				
15. Features for Operational Excellence	Define Key Terms				
15. Features for Operational Excellence	Complete Q&A section				
Practice Test	Take practice test in study mode using Exam Bank 1 questions for Chapter 15 in practice test software				
16. Final Preparation					
16. Final Preparation	Take practice test in study mode for all Book Questions in practice test software				

16. Final Preparation	Review all Key Topics in all chapters				
16. Final Preparation	Complete all Memory Tables/Lists in Appendix B				
16. Final Preparation	Take practice test in practice exam mode using Exam Bank #1 questions for all chapters				
16. Final Preparation	Take practice test in practice exam mode using Exam Bank #2 questions for all chapters				

Glossary

AKP Asynchronous Key Prefetch; improves Aurora performance by anticipating the rows needed to run queries in which a JOIN query requires use of the BKA Join algorithm and MRR optimization features **Amazon Redshift Advisor** An AWS tool that provides specific recommendations to increase performance and reduce costs associated with Amazon Redshift **Archive** Data that is stored in AWS Glacier

Asynchronous decoupling An approach to decoupled components in which the components do not need to be present at all times for the system to function **Aurora** A MySQL-and PostgreSQL-compatible relational database engine that combines the speed and availability of high-end commercial databases with the simplicity and cost-effectiveness of open-source databases **Auto Scaling** A feature that helps you maintain application availability and allows you to scale your Amazon EC2 capacity automatically according to conditions that you define **AWS Batch** A service for efficiently running batch jobs against AWS resources

AWS Cloud Compliance A set of online tools to assist you in achieving regulatory compliance with your AWS solutions

AWS Config A service that enables you to assess, audit, and evaluate the configurations of your AWS resources **AWS**

Global Infrastructure Regions and Availability Zones located around the globe

AWS KMS The encryption management service for AWS

AWS Simple Monthly Calculator An online tool that allows you to estimate monthly AWS charges **AWS Storage**

Gateway A service that seamlessly enables hybrid storage between on-premises storage environments and the AWS Cloud **AWS X-Ray** A tool for tracking execution of code in distributed applications

Billing metrics A set of metrics in CloudWatch that permit the careful monitoring of AWS costs; you can also set alarms based on this information **Bucket** Logical storage container in S3

Bursting Throughput Mode Throughput scales as the file system grows

CapEx Capital expenditures

ClassicLink A feature that allows you to connect EC2-Classic instances to your VPC

CloudFormation A service that gives developers and system administrators an easy way to create and manage a collection of related AWS resources, provisioning and updating them in an orderly and predictable fashion **CloudFront** A global

content delivery network (CDN) service that accelerates delivery of your websites, APIs, video content, or other web assets **CloudTrail** A web service that records AWS API calls for your account and delivers log files to you **CloudWatch** A monitoring service for AWS Cloud resources and the applications you run on AWS

Convertible RI A reserved instance that permits modifications from the original reservation **Cooldown period** Mandatory waiting period between Auto Scaling actions

Cost Explorer A tool in the Management Console that permits you to create detailed reports based on your AWS costs **Data tier** The storage media for the data required by an application

Database Migration Service A service that helps you migrate databases into or out of AWS easily and securely **DAX** DynamoDB Accelerator; a caching engine specially designed for DynamoDB

Decoupling Creating components that have the capability to be autonomous from each other to some degree **DHCP option sets** Components that allow you to provide settings such as a DNS name or DNS server address to instances **Direct Connect** A tool that makes it easy to establish a dedicated network connection from your premises to AWS

Directory Services A service that enables your directory-aware workloads and AWS resources to use managed Active Directory in the AWS Cloud **Distribution** A group of resources cached and shared by CloudFront

DNS Domain Name System; the DNS server provided by AWS or the one or more servers that you provide **Durability** Resiliency; in data storage, this term means the chance of data loss

DynamoDB A fast and flexible NoSQL database service for all applications that need consistent, single-digit millisecond latency at any scale **EBS-optimized instance** Dedicated bandwidth between EC2 and EBS

Edge location A special data center designed for delivering CloudFront cached resources

Egress-only Internet gateway A VPC component that provides Internet access for IPv6 addressed instances **Elastic Beanstalk** An easy-to-use service for deploying and scaling web applications and services **Elastic Block Store** A resource that provides persistent block storage volumes for use with EC2 instances in the AWS Cloud **Elastic Compute Cloud (EC2)** Virtual machine compute resources in AWS

Elastic Container Registry A fully-managed Docker container registry in AWS

Elastic Container Service A scalable, high-performance container management service that supports Docker containers **Elastic Container Service for Kubernetes** A fully managed service that makes it easy for you to use Kubernetes on AWS without having to be an expert in managing Kubernetes

clusters **Elastic File System** A service that provides simple, scalable file storage for use with Amazon EC2 instances in the AWS Cloud **Elastic IP addresses** Public IP addresses that you can assign flexibly in your AWS VPC

Elastic Load Balancing A feature that automatically distributes incoming application traffic across multiple EC2 instances **ElastiCache** A web service that makes it easy to deploy, operate, and scale an in-memory cache in the cloud **Elasticity** The ability of resources to scale up and down

Fargate A technology for Amazon ECS and EKS that allows you to run containers without having to manage servers or clusters **FT** Fault tolerance; a subset method of high availability designed to create a zero downtime solution and zero performance degradation due to a component failure **Glacier** A secure, durable, and extremely low-cost storage

service for data archiving and long-term backup **Global secondary index** An index with a partition key and a sort key that can be different from those on the base table **Greengrass** An AWS service that allows IoT devices to run code that is synched from the cloud **Groups** Objects that contain user accounts; permissions can be assigned to groups and not users to implement scalability in the design **HA** High availability; a goal in solution design that focuses on automatic failover and recoverability from an issue in the solution **HTTPS** The secure version of HTTP for protecting data in transit

Identity and Access Management (IAM) A tool that enables you to securely control access to AWS services and resources for your users **Initialization** A process in which each block of an EBS volume is accessed in advance of deployment; it is used when you have restored a volume from a Snapshot **Instance store** Legacy ephemeral storage option for EC2 instances

Internet gateway Virtual device that provides Internet access to instances

Key Management Service (KMS) A managed service that makes it easy for you to create and control the encryption keys used to encrypt your data **Kinesis** A data platform that makes it easy to collect, process, and analyze real-time, streaming data **Lambda** A serverless compute service of AWS that permits the execution of custom code or functions without the provisioning of virtual servers **Latency** Delay (from the storage to the instance, for example)

Lazy loading Caching data strictly based on requests for the data

Lightsail An interface that enables you to launch and manage a virtual private server with AWS

Local secondary index An index that has the same partition key as the base table but a different sort key **Logic tier** The server that contains the code providing the functionality of the application or technology **Loose decoupling** Creating decoupled components that have few dependencies on each other

memcached A simple caching engine supported by ElastiCache

Multi-AZ The reference to using multiple Availability Zones in an AWS design; often done to ensure high availability (HA)

Multi-Tier architecture The use of multiple systems or platforms to implement an application or technology **NAT** Network Address Translation as supported in AWS

Network ACL A security mechanism in AWS used to control traffic flow in and out of subnets within VPCs **Network**

interface Virtual interface that represents a NIC

NFS Network File System; a standard shared file system protocol

NIST The National Institute of Standards and Technology, which provides excellent guidance on best practices **Object storage** The data storage technique of AWS S3

OpEx Operating expenditures

OpsWorks A configuration management service that uses Chef, an automation platform that treats server configurations as code **Patch Manager** A component of AWS Systems

Manager that automates the deployment of patches to

operating systems in your AWS infrastructure **Policy** An object in IAM that defines the actual permissions assigned to a resource or service in AWS

Presentation tier Components that users interact with for a user interface

Provisioned Throughput Mode An operating mode available for high throughput needs or requirements greater than Bursting Throughput mode allows **Redis** A cache engine supported by ElastiCache that supports complex data types and multi-AZ deployments **Redshift** A fast, fully managed, petabyte-scale data warehouse that makes it simple and cost-effective to analyze all your data using your existing business intelligence tools **Relational Database Service (RDS)** A service that makes it easy to set up, operate, and scale a relational database in the cloud **Reserved instances** AWS capacity utilized at a discount over default on-demand pricing as a result of the reservation **Role** Similar to a user but does not represent one user in the organization; instead, it is intended to be assumable by anyone who needs it **Root user** Role created when you create your AWS account; this role has unrestricted access to all aspects of your account **Route 53** A highly available and scalable cloud Domain Name System (DNS) web service

Route table Routing instructions for your subnets

RPO Recovery point objective; the point to which data must be restored

RTO Recovery time objective; the amount of time in which a recovery must be completed

Scheduled RI Reserved instances that will operate within a certain time window

Security group A security component in AWS used to secure EC2 and other resources

Serverless Application Repository A service that enables you to deploy code samples, components, and complete applications for common use cases such as web and mobile back ends, event and data processing, logging, monitoring, IoT, and more **Simple Notification Service (SNS)** A fast, flexible, distributed, and fully managed push notification service that lets you send individual messages or to distribute messages to many recipients **Simple Queue Service (SQS)** A fast, reliable, scalable, distributed, and fully managed message queuing service **Simple Storage Service (S3)** Object storage with a simple web service interface to store and retrieve any amount of data from anywhere on the web **Single-Tier**

Architecture A type of architecture in which all required components for a software application or technology are provided on a single server or platform **Snowball** A petabyte-scale data transport solution that uses secure appliances to transfer large amounts of data into and out of AWS

Standard RI A standard reserved instance; this type provides the greatest potential cost savings **Storage classes** Different classes of S3 storage used to balance performance and costs

Synchronous decoupling An approach to decoupling that means decoupled components must be present for the system to function properly **TCP Selective Acknowledgment** A performant S3 service that improves recovery time after large numbers of packet loss **TCP Window Scaling** A performant

S3 service that supports window sizes larger than 64 KB

Technical Account Managers (TAMs) Experts that are available to you if you have an Enterprise Support level of service
Trusted Advisor An online resource to help you reduce cost, increase performance, and improve security by optimizing your AWS environment
TTL The time to live value in Route 53 that dictates the amount of time resource record information cached by DNS resolvers
Users Components in IAM that represent users in your organization

Vault Logical storage structure in AWS Glacier

Versioning Keeping multiple copies of objects in AWS S3 for various time points

Virtual Private Cloud (VPC) Virtual network components in AWS

Volume queue length The number of pending I/O requests for a device

VPC endpoints Devices that allow you to privately connect your VPC to supported endpoint services
VPC peering A connection that permits communications between VPCs

Web Application Firewall (WAF) A feature that helps protect your web applications from common web exploits that could affect application availability, compromise security, or consume excessive resources
Write through A caching strategy designed to minimize stale cache data