

**Certified Solutions Architect Official**

**Study Guide - Associate Exam**

****

**Joe Baron, Hisham Baz, Tim Bixler, Biff Gaut, Kevin E. Kelly, Sean Senior, John Stamper**

****

Senior Acquisitions Editor: Kenyon Brown

Project Editor: Gary Schwartz

Production Editor: Dassi Zeidel

Copy Editor: Kezia Endsley

Editorial Manager: Mary Beth Wakefield

Production Manager: Kathleen Wisor

Executive Editor: Jim Minatel

Book Designers: Judy Fung and Bill Gibson

Proofreader: Nancy Carrasco

Indexer: Johnna vanHoose Dinse

Project Coordinator, Cover: Brent Savage

Cover Designer: Wiley

Cover Image: ©Getty Images, Inc./Jeremy Woodhouse

Copyright © 2017 by AWS

Published by John Wiley & Sons, Inc. Indianapolis, Indiana

Published simultaneously in Canada

ISBN: 978-1-119-13855-6

ISBN: 978-1-119-13955-3 (ebk.)

ISBN: 978-1-119-13954-6 (ebk.)

Manufactured in the United States of America

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at http://www.wiley.com/go/permissions.

Limit of Liability/Disclaimer of Warranty: The publisher and the author make no representations or warranties with respect to the accuracy or completeness of the contents of this work and specifically disclaim all warranties, including without limitation warranties of fitness for a particular purpose. No warranty may be created or extended by sales or promotional materials. The advice and strategies contained herein may not be suitable for every situation. This work is sold with the understanding that the publisher is not engaged in rendering legal, accounting, or other professional services. If professional assistance is required, the services of a competent professional person should be sought. Neither the publisher nor the author shall be liable for damages arising herefrom. The fact that an organization or Web site is referred to in this work as a citation and/or a potential source of further information does not mean that the author or the publisher endorses the information the organization or Web site may provide or recommendations it may make. Further, readers should be aware that Internet Web sites listed in this work may have changed or disappeared between when this work was written and when it is read.

For general information on our other products and services or to obtain technical support, please contact our Customer Care Department within the U.S. at (877) 762-2974, outside the U.S. at (317) 572-3993 or fax (317) 572-4002.

Wiley publishes in a variety of print and electronic formats and by print-on-demand. Some material included with standard print versions of this book may not be included in e-books or in print-on-demand. If this book refers to media such as a CD or DVD that is not included in the version you purchased, you may download this material at http://booksupport.wiley.com. For more information about Wiley products, visit www.wiley.com.

Library of Congress Control Number: 2016949703

TRADEMARKS: Wiley, the Wiley logo, and the Sybex logo are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates, in the United States and other countries, and may not be used without written permission. AWS is a registered trademark of Amazon Technologies, Inc. All other trademarks are the property of their respective owners. John Wiley & Sons, Inc. is not associated with any product or vendor mentioned in this book.

*For the original AWS instructor, Mike Culver, who taught us how to teach, lead, and inspire with tenacity and kindness.*

**CONTENTS**

Acknowledgments

About the Authors

Foreword

Introduction

Assessment Test

Answers to Assessment Test

Chapter 1 Introduction to AWS

What Is Cloud Computing?

AWS Fundamentals

AWS Cloud Computing Platform

Summary

Exam Essentials

Review Questions

Chapter 2 Amazon Simple Storage Service (Amazon S3) and Amazon Glacier Storage Introduction

Object Storage versus Traditional Block and File Storage

Amazon Simple Storage Service (Amazon S3) Basics

Buckets

Amazon S3 Advanced Features

Amazon Glacier

Summary

Exam Essentials

Exercises

Review Questions

Chapter 3 Amazon Elastic Compute Cloud (Amazon EC2) and Amazon Elastic Block Store (Amazon EBS)

Introduction

Amazon Elastic Compute Cloud (Amazon EC2)

Amazon Elastic Block Store (Amazon EBS)

Summary

Exam Essentials

Exercises

Review Questions

Chapter 4 Amazon Virtual Private Cloud (Amazon VPC)

Introduction

Amazon Virtual Private Cloud (Amazon VPC)

Subnets

Route Tables

Internet Gateways

Dynamic Host Configuration Protocol (DHCP) Option Sets

Elastic IP Addresses (EIPs)

Elastic Network Interfaces (ENIs)

Endpoints

Peering

Security Groups

Network Access Control Lists (ACLs)

Network Address Translation (NAT) Instances and NAT Gateways Virtual Private Gateways (VPGs), Customer Gateways (CGWs), and Virtual Private Networks (VPNs)

Summary

Exam Essentials

Exercises

Review Questions

Chapter 5 Elastic Load Balancing, Amazon CloudWatch, and Auto Scaling Introduction

Elastic Load Balancing

Amazon CloudWatch

Auto Scaling

Summary

Exam Essentials

Exercises

Review Questions

Chapter 6 AWS Identity and Access Management (IAM)

Principals

Authentication

Authorization

Other Key Features

Summary

Exam Essentials

Exercises

Review Questions

Chapter 7 Databases and AWS

Database Primer

Amazon Relational Database Service (Amazon RDS) Amazon Redshift

Amazon DynamoDB

Summary

Exam Essentials

Exercises

Review Questions

Chapter 8 SQS, SWF, and SNS

Amazon Simple Queue Service (Amazon SQS)

Amazon Simple Workflow Service (Amazon SWF) Amazon Simple Notification Service (Amazon SNS) Summary

Exam Essentials

Exercises

Review Questions

Chapter 9 Domain Name System (DNS) and Amazon Route 53 Domain Name System (DNS)

Amazon Route 53 Overview

Summary

Exam Essentials

Exercises

Review Questions

Chapter 10 Amazon ElastiCache

Introduction

In-Memory Caching

Amazon ElastiCache

Summary

Exam Essentials

Exercises

Review Questions

Chapter 11 Additional Key Services

Introduction

Storage and Content Delivery

Security

Analytics

DevOps

Summary

Exam Essentials

Review Questions

Chapter 12 Security on AWS

Introduction

Shared Responsibility Model

AWS Compliance Program

AWS Global Infrastructure Security

AWS Account Security Features

AWS Cloud Service-Specific Security

Summary

Exam Essentials

Exercises

Review Questions

Chapter 13 AWS Risk and Compliance

Introduction

Overview of Compliance in AWS

Evaluating and Integrating AWS Controls

AWS Risk and Compliance Program

AWS Reports, Certifications, and Third-Party Attestations Summary

Exam Essentials

Review Questions

Chapter 14 Architecture Best Practices

Introduction

Design for Failure and Nothing Fails

Implement Elasticity

Leverage Different Storage Options

Build Security in Every Layer

Think Parallel

Loose Coupling Sets You Free

Don’t Fear Constraints

Summary

Exam Essentials

Exercises

Review Questions

Appendix A Answers to Review Questions

Chapter 1: Introduction to AWS

Chapter 2: Amazon Simple Storage Service (Amazon S3) and Amazon Glacier Storage

Chapter 3: Amazon Elastic Compute Cloud (Amazon EC2) and Amazon Elastic Block Store (Amazon EBS)

Chapter 4: Amazon Virtual Private Cloud (Amazon VPC)

Chapter 5: Elastic Load Balancing, Amazon CloudWatch, and Auto Scaling Chapter 6: AWS Identity and Access Management (IAM)

Chapter 7: Databases and AWS

Chapter 8: SQS, SWF, and SNS

Chapter 9: Domain Name System (DNS) and Amazon Route 53

Chapter 10: Amazon ElastiCache

Chapter 11: Additional Key Services

Chapter 12: Security on AWS

Chapter 13: AWS Risk and Compliance

Chapter 14: Architecture Best Practices

Advert

EULA

**List of Tables** Chapter 3

TABLE 3.1

TABLE 3.2

TABLE 3.3

TABLE 3.4

TABLE 3.5

TABLE 3.6

Chapter 4

TABLE 4.1

TABLE 4.2

TABLE 4.3

TABLE 4.4

TABLE 4.5

Chapter 6

TABLE 6.1

TABLE 6.2

TABLE 6.3

Chapter 7

TABLE 7.1

TABLE 7.2

TABLE 7.3

TABLE 7.4

TABLE 7.5

Chapter 12

TABLE 12.1

Chapter 14

TABLE 14.1

**List of Illustrations**

Chapter 1

**FIGURE 1.1** Six advantages of cloud computing

**FIGURE 1.2** AWS Cloud computing platform

**FIGURE 1.3** Auto scaling capacity

**FIGURE 1.4** AWS CloudFormation workflow summary

Chapter 3

**FIGURE 3.1** Memory and vCPUs for the m4 instance family

**FIGURE 3.2** A workload using a mix of On-Demand and Reserved Instances Chapter 4

**FIGURE 4.1** VPC, subnets, and a route table

**FIGURE 4.2** VPC, subnet, route table, and an Internet gateway

**FIGURE 4.3** VPC peering connections do not support transitive routing **FIGURE 4.4** VPC with VPN connection to a customer network

Chapter 5

**FIGURE 5.1** Auto Scaling group behind an Elastic Load Balancing load balancer **FIGURE 5.2** Auto Scaling group with policy

**FIGURE 5.3** Amazon CloudWatch alarm triggering scaling out

Chapter 6

**FIGURE 6.1** Different identities authenticating with AWS

**FIGURE 6.2** Associating IAM users with policies

Chapter 7

**FIGURE 7.1** Multi-AZ Amazon RDS architecture

**FIGURE 7.2** Amazon Redshift cluster architecture

**FIGURE 7.3** Table, items, attributes relationship

**FIGURE 7.4** Table partitioning

Chapter 8

**FIGURE 8.1** Message lifecycle

**FIGURE 8.2** Diagram of visibility timeout

**FIGURE 8.3** Amazon SWF workflow illustration

**FIGURE 8.4** Diagram of topic delivery

**FIGURE 8.5** Diagram of fanout scenario

Chapter 9

**FIGURE 9.1** FQDN components

Chapter 10

**FIGURE 10.1** Common caching architecture

**FIGURE 10.2** Redis replication group

Chapter 11

**FIGURE 11.1** Delivering static and dynamic content

**FIGURE 11.2** High availability CloudHSM architecture

**FIGURE 11.3** Amazon Kinesis Firehose

**FIGURE 11.4** Amazon Kinesis Streams

**FIGURE 11.5** Example pipeline

**FIGURE 11.6** Simple application server stack

**FIGURE 11.7** Simple application server stack with AWS OpsWorks **FIGURE 11.8** Creating a stack workflow

**FIGURE 11.9** Updating a stack workflow

**FIGURE 11.10** AWS Trusted Advisor Console dashboard

Chapter 12

**FIGURE 12.1** The shared responsibility model

**FIGURE 12.2** Amazon Web Services regions

**FIGURE 12.3** Amazon EC2 multiple layers of security

**FIGURE 12.4** Amazon EC2 security group firewall

**FIGURE 12.5** Amazon VPC network architecture

**FIGURE 12.6** Flexible network architectures

Chapter 13

**FIGURE 13.1** Shared responsibility model

Chapter 14

**FIGURE 14.1** Simple web application architecture

**FIGURE 14.2** Updated web application architecture with redundancy **FIGURE 14.3** Updated web application architecture with auto scaling

**FIGURE 14.4** Updated web application architecture with Amazon S3 and Amazon CloudFront

**FIGURE 14.5** Updated web application architecture with Amazon ElastiCache and Amazon DynamoDB

**FIGURE 14.6** Tight and loose coupling

**FIGURE 14.7** Sample web application for chapter exercises

**Acknowledgments**

The authors would like to thank a few people who helped us develop and write this *AWS Certified Solutions Architect Official Study Guide: Associate Exam.*

First, thanks to all our families who put up with us spending weekends and evenings creating content, writing questions, and reviewing each other's chapters. Their patience and support made this book possible.

Niamh O'Byrne, AWS Certification Manager, who introduced all of the authors and many more solutions architects at AWS to certification testing and got this book started by challenging some of us to extend our reach and help more cloud practitioners get certified.

Nathan Bower and Victoria Steidel, amazing technical writers at AWS who reviewed and edited all the content and every question and gently made us better writers and communicators. They were tireless in reviewing and helping us hone and focus our content.

Patrick Shumate, a fellow AWS solutions architect who contributed test questions right when we needed the help to get us over the finish line.

We could not have written this book without the help of our friends at Wiley. Kenyon Brown, Senior Acquisitions Editor, corralled us and focused us on the end goal. Additionally, we were guided by Gary Schwartz, Project Editor; Kezia Endsley, Copyeditor; and Dassi Zeidel, Production Editor who took output from different authors and turned it into a cohesive and complete finished product.

Lastly, we want to thank all the solutions architects at AWS who participated in certification blueprint development, question writing, and review sessions, and the development of a world-class certification program for cloud practitioners that is setting the standard for our industry.

**About the Authors**

****

**Joe Baron**, Principal Solutions Architect for AWS, is currently working with customers in the Southeastern United States. Joe joined AWS in 2009 as one of the first solutions architects, and in the years since he has helped customers of all sizes, from small startups to some of the largest enterprises in the world, to architect their infrastructures and migrate their applications to the cloud. He was also an early contributor to the AWS Associate and Professional Certified Solutions Architect programs. Joe holds a BS degree in engineering physics from Cornell University and is proud to be an “expert generalist.” Prior to joining AWS, Joe had 25 years of experience in technology, with roles in data center automation, virtualization, life sciences, high-performance computing, 3D visualization, hardware and software development, and Independent Software Vendor (ISV) program management. He is also a dedicated husband to Carol and father of two children, Matt and Jessie. When not helping customers migrate all the things to the cloud, Joe is an amateur classical pianist and collector of traditional woodworking tools. He lives in the Raleigh, NC area.



**Hisham Baz** is a passionate software engineer and systems architect with expertise building distributed applications and high-performance, mission-critical systems. Since 2013, Hisham has been a solutions architect with AWS working with customers like Pinterest, Airbnb, and General Electric to build resilient architectures in the cloud with a focus on big data and analytics. Prior to Amazon, Hisham founded two early-stage startups, modernized the communications network connecting critical transportation infrastructure, and improved cellular networks with large-scale data analytics. Hisham is based in San Francisco, CA and lives with his wife, Suki. They can often be found hiking the redwoods.



**Tim Bixler**, Commercial Americas Southeast Area Solutions Architecture Leader for AWS, leads teams of solutions architects who provide AWS technical enablement, evangelism, and knowledge transfer to customers like Capital One, The Coca-Cola Company, AOL, Koch Industries, Cox Automotive, NASCAR, Emdeon, and Neustar. Tim has over 20 years of experience in improving systems and operational performance, productivity, and customer satisfaction for private and public global corporations as well as government agencies. He is also a public speaker for Amazon and enjoys helping customers adopt innovative solutions on AWS. But if you ask his 7-year-old son TJ what he does, he might say that daddy is a builder and a fixer. When not otherwise tasked, you can find him burrowed in his lab building robots driven by microcontrollers or at the local BrickFair admiring the creations that he has no time to build.



**Biff Gaut** started writing programs for a living on CP/M on the Osborne 1. Since those early days, he obtained a BS in engineering from Virginia Tech while writing C code on MS-DOS, married his wife, Holly, while writing his first GUI apps, and raised two children while transitioning from COM objects in C++ to web apps in .NET. Along the way, he led development teams from 1 to 50 members for companies including NASDAQ, Thomson Reuters, Verizon, Microsoft, FINRA, and Marriott. He has collaborated on two books and spoken at countless conferences, including Windows World and the Microsoft PDC. Biff is currently a solutions architect at AWS, helping customers across the country realize the benefits of the cloud by deploying secure, available, efficient workloads on AWS. And yes, that’s his real name.



**Kevin E. Kelly**, Solutions Architecture Manager and early contributor to the AWS Solutions Architecture Certification exams, has been at AWS for over seven years helping companies architect their infrastructures and migrate their applications to the cloud. Kevin has a BS in computer science from Mercer University and a Master of Information Systems in business from the University of Montana. Before joining Amazon, Kevin was an Air Force officer, a programmer—including embedded programming—and a technical presales leader. Kevin has been the chairman of the Worldwide Web Consortium (W3C) Compound Document Format Working Group and led that open-standards working group in developing the Web Interactive Compound Document (WICD) profile for mobile and desktop devices. He has also served as the W3C Advisory Council Representative for Health Level 7 (HL7). Kevin lives in Virginia with his wife, Laurie, and their two daughters, Caroline and Amelia. Kevin is an amateur violin and mandolin player and a zymurgist.



**Sean Senior** is a solutions architect at AWS. Sean is a builder at heart and thrives in a fast paced environment with continuous challenges. Sean has a BS in computer information and sciences from the University of Maryland University College. Sean is a devoted husband and

father of a beautiful girl. He is a U.S. Navy veteran, avid sports fan, and gym rat. He loathes talking about himself in the third person, but can be persuaded to do so for a good reason.



**John Stamper**, Principal Solutions Architect at AWS, is a co-inventor for multiple AWS patents and is particularly fond of distributed systems at scale. John holds a BS in mathematics from James Madison University (94) and an MS in Information Systems from George Mason University (04). In addition to building systems on the cloud and helping customers reimagine their businesses, John is a dedicated husband and father of three children. He is a CrossFit athlete, youth sports coach, and vocal supporter of the arts.

**Foreword**

This *AWS Certified Solutions Architect Official Study Guide: Associate Exam* has been written to help you prepare for the AWS Certified Solutions Architect – Associate exam. This certification is becoming an increasingly important credential that every information technology professional and cloud practitioner who plans, designs, and builds application architectures for deployment on AWS should obtain. Passing the AWS Certified Solutions Architect – Associate exam demonstrates to your colleagues, employers, and the industry at large that you know how to build and deploy AWS solutions that are highly available, secure, performant, and cost effective.

This study guide was written by AWS solutions architects who wrote and reviewed exam questions for the AWS Certified Solutions Architect exams. Although nothing replaces hands on experience building and deploying a variety of cloud applications and controls on AWS, this study guide, and the questions and exercises in each chapter, provide you with coverage of the basic AWS Cloud services combined with architectural recommendations and best practices that will help prepare you for the exam. Combining this study guide with production application deployment experience and taking the practice exams online will prepare you well and allow you to take the exam with confidence. Adding the AWS Certified Solutions Architect—Associate certification to your credentials will establish you as an industry recognized solutions architect for the AWS platform!

—Kevin E. Kelly

Americas Solutions Architecture Lead

AWS Certified Solutions Architect – Associate

AWS Certified Solutions Architect – Professional

Herndon, VA

**Introduction**

Studying for any certification exam can seem daunting. This *AWS Certified Solutions Architect Official Study Guide: Associate Exam* was designed and developed with relevant topics, questions, and exercises to enable a cloud practitioner to focus their precious study time and effort on the germane set of topics targeted at the right level of abstraction so they can confidently take the AWS Certified Solutions Architect – Associate exam.

This study guide presents a set of topics needed to round out a cloud practitioner’s hands-on experiences with AWS by covering the basic AWS Cloud services and concepts within the scope of the AWS Certified Solutions Architect – Associate exam. This study guide begins with an introduction to AWS, which is then followed by chapters on specific AWS Cloud services. In addition to the services chapters, the topics of security, risk and compliance, and architecture best practices are covered, providing the reader with a solid base for understanding how to build and deploy applications on the AWS platform. Furthermore, the AWS architectural best practices and principles are reinforced in every chapter and reflected in the self-study questions and examples to highlight the development and deployment of applications for AWS that are secure, highly available, performant, and cost effective. Each chapter includes specific information on the service or topic covered, followed by an Exam Essentials section that contains key information needed in your exam preparation. The Exam Essentials section is followed by an Exercise section with exercises designed to help reinforce the topic of the chapter with hands-on learning. Next, each chapter contains sample questions to get you accustomed to answering questions about AWS Cloud services and architecture topics. The book also contains a self-assessment exam with 25 questions, two practice exams, with 50 questions each to help you gauge your readiness to take the exam, and flashcards to help you learn and retain key facts needed to prepare for the exam.

If you are looking for a targeted book written by solutions architects who wrote, reviewed, and developed the AWS Certified Solutions Architect – Associate exam, then this is the book for you.

**What Does This Book Cover?**

This book covers topics you need to know to prepare for the Amazon Web Services (AWS) Certified Solutions Architect – Associate exam:

**Chapter 1: Introduction to AWS** This chapter provides an introduction to the AWS Cloud computing platform. It discusses the advantages of cloud computing and the fundamentals of AWS. It provides an overview of the AWS Cloud services that are fundamentally important for the exam.

**Chapter 2: Amazon Simple Storage Service (Amazon S3) and Amazon Glacier Storage** This chapter provides you with a basic understanding of the core object storage services available on AWS: Amazon Simple Storage Service (Amazon S3) and Amazon Glacier. These services are used to store objects on AWS.

**Chapter 3: Amazon Elastic Compute Cloud (Amazon EC2) and Amazon Elastic Block Store (Amazon EBS)** In this chapter, you will learn how Amazon Elastic Compute Cloud (Amazon EC2) and Amazon Elastic Block Store (Amazon EBS) provide the basic elements of compute and block-level storage to run your workloads on AWS.

**Chapter 4: Amazon Virtual Private Cloud (Amazon VPC)** This chapter describes Amazon Virtual Private Cloud (Amazon VPC), which is a custom-defined virtual network within AWS. You will learn how to design secure architectures using Amazon VPC to provision your own logically isolated section of AWS.

**Chapter 5: Elastic Load Balancing, Amazon CloudWatch, and Auto Scaling** In this chapter, you will learn how Elastic Load Balancing, Amazon CloudWatch, and Auto Scaling work independently and together to help you efficiently and cost-effectively deploy highly available and optimized workloads on AWS.

**Chapter 6: AWS Identity and Access Management (IAM)** This chapter covers AWS Identity and Access Management (IAM), which is used to secure transactions with the AWS resources in your AWS account.

**Chapter 7: Databases and AWS** This chapter covers essential database concepts and introduces three of AWS managed database services: Amazon Relational Database Service (Amazon RDS), Amazon DynamoDB, and Amazon Redshift. These managed services simplify the setup and operation of relational databases, NoSQL databases, and data warehouses.

**Chapter 8: SQS, SWF, and SNS** This chapter focuses on application services in AWS, specifically Amazon Simple Queue Service (Amazon SQS), Amazon Simple Workflow Service (SWF), and Amazon Simple Notification Service (Amazon SNS). It also covers architectural guidance on using these services and the use of Amazon SNS in mobile applications.

**Chapter 9: Domain Name System (DNS) and Amazon Route 53** In this chapter, you will learn about Domain Name System (DNS) and the Amazon Route 53 service, which is designed to help users find your website or application over the Internet.

**Chapter 10: Amazon ElastiCache** This chapter focuses on building high-performance applications using in-memory caching technologies and Amazon ElastiCache.

**Chapter 11: Additional Key Services** Additional services not covered in other chapters are

covered in this chapter. Topics include Amazon CloudFront, AWS Storage Gateway, AWS Directory Service, AWS Key Management Service (KMS), AWS CloudHSM, AWS CloudTrail, Amazon Kinesis, Amazon Elastic Map Reduce (Amazon EMR), AWS Data Pipeline, AWS Import/Export, AWS OpsWorks, AWS CloudFormation, AWS Elastic Beanstalk, AWS Trusted Advisor, and AWS Config.

**Chapter 12: Security on AWS** This chapter covers the relevant security topics that are within scope for the AWS Certified Solutions Architect – Associate exam.

**Chapter 13: AWS Risk and Compliance** This chapter covers topics associated with risk and compliance, risk mitigation, and the shared responsibility model of using AWS.

**Chapter 14: Architecture Best Practices** The final chapter covers the AWS recommended design principles and best practices for architecting systems and applications for the Cloud.

**Interactive Online Learning Environment and Test Bank**

The authors have worked hard to provide some really great tools to help you with your certification process. The interactive online learning environment that accompanies the *AWS Certified Solutions Architect Official Study Guide: Associate Exam* provides a test bank with study tools to help you prepare for the certification exam—and increase your chances of passing it the first time! The test bank includes the following:

**Sample Tests** All the questions in this book are provided, including the assessment test at the end of this Introduction and the chapter tests that include the review questions at the end of each chapter. In addition, there are two practice exams with 50 questions each. Use these questions to test your knowledge of the study guide material. The online test bank runs on multiple devices.

**Flashcards** The online text banks include 100 flashcards specifically written to hit you hard, so don’t get discouraged if you don’t ace your way through them at first. They’re there to ensure that you’re really ready for the exam. And no worries—armed with the review questions, practice exams, and flashcards, you’ll be more than prepared when exam day comes. Questions are provided in digital flashcard format (a question followed by a single correct answer). You can use the flashcards to reinforce your learning and provide last minute test prep before the exam.

**Glossary** A glossary of key terms from this book is available as a fully searchable PDF.

Go to http://www.wiley.com/go/sybextestprep to register and gain access to this interactive online learning environment and test bank with study tools.

**Exam Objectives**

The AWS Certified Solutions Architect—Associate exam is intended for people who have experience in designing distributed applications and systems on the AWS platform. Here are some of the key exam topics that you should understand for this exam:

Designing and deploying scalable, highly available, and fault-tolerant systems on AWS Migrating existing on-premises applications to AWS

Ingress and egress of data to and from AWS

Selecting the appropriate AWS service based on data, compute, database, or security requirements

Identifying appropriate use of AWS architectural best practices

Estimating AWS costs and identifying cost control mechanisms

In general, candidates should have the following:

One or more years of hands-on experience designing highly available, cost efficient, secure, fault tolerant, and scalable distributed systems on AWS

In-depth knowledge of at least one high-level programming language

Ability to identify and define requirements for an AWS-based application Experience with deploying hybrid systems with on-premises and AWS components

Capability to provide best practices for building secure and reliable applications on the AWS platform

The exam covers four different domains, with each domain broken down into objectives and subobjectives.

**Objective Map**

The following table lists each domain and its weighting in the exam, along with the chapters in the book where that domain’s objectives and subobjectives are covered.

|  |  |  |
| --- | --- | --- |
| **Domain** | **Percentageof Exam** | **Chapter** |
| 1 Domain 1.0: Designing highly available, cost-efficient, fault tolerant, scalable systems | 60% |  |
| 1.1 Identify and recognize cloud architecture considerations, such as fundamental components and effective designs. |  | 1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 14 |
| Content may include the following: |  |  |
| How to design cloud services |  | 1, 2, 3, 4, 8, 9, 11, 14 |

Planning and design 1, 2, 3, 4, 7, 8, 9,

10, 11, 14

|  |  |  |
| --- | --- | --- |
| Monitoring and logging |  | 2, 3, 8, 9, 11 |
| Familiarity with: |  |  |
| Best practices for AWS architecture |  | 1, 2, 4, 7, 8, 9, 10, 14 |
| Developing to client specifications, including pricing/cost (e.g., on Demand vs. Reserved vs. Spot; RTO and RPO DR Design) |  | 2, 7, 9 |
| Architectural trade-off decisions (e.g., high availability vs. cost, Amazon Relational Database Service (RDS) vs. installing your own database on Amazon Elastic Compute Cloud (EC2)) |  | 2, 4, 7, 8, 9, 10 |
| Hybrid IT architectures (e.g., Direct Connect, Storage Gateway, VPC, Directory Services) |  | 1, 2, 4, 14 |
| Elasticity and scalability (e.g., Auto Scaling, SQS, ELB, CloudFront) |  | 1, 2, 5, 7, 8, 9, 10, 14 |
| 2 Domain 2.0: Implementation/Deployment | 10% |  |
| 2.1 Identify the appropriate techniques and methods using Amazon EC2, Amazon S3, AWS Elastic Beanstalk, AWS CloudFormation, AWS OpsWorks, Amazon Virtual Private Cloud (VPC), and AWS Identity and Access Management (IAM) to code and implement a cloud solution. |  | 1, 2, 3, 4, 5, 6, 8,  11, 13 |
| Content may include the following: |  |  |
| Configure an Amazon Machine Image (AMI). |  | 2, 3, 11 |
| Operate and extend service management in a hybrid IT architecture. |  | 1, 4 |
| Configure services to support compliance requirements in the cloud. |  | 2, 3, 4,  11, 13 |
| Launch instances across the AWS global infrastructure. |  | 1, 2, 3, 5, 8, 11 |
| Configure IAM policies and best practices. |  | 2, 6 |
| 3 Domain 3.0: Data Security | 20% |  |
| 3.1 Recognize and implement secure practices for optimum cloud deployment and maintenance. |  | 2, 4, 10, 12, 13 |
| Content may include the following: |  |  |
| AWS shared responsibility model |  | 12, 13 |
| AWS platform compliance |  | 11, 12, 13 |

AWS security attributes (customer workloads down to physical layer) 4, 11, 12,

13

|  |  |  |
| --- | --- | --- |
| AWS administration and security services |  | 7, 10, 11, 12 |
| AWS Identity and Access Management (IAM) |  | 6, 12 |
| Amazon Virtual Private Cloud (VPC) |  | 4, 12 |
| AWS CloudTrail |  | 11, 12 |
| Ingress vs. egress filtering, and which AWS services and features fit |  | 11, 12 |
| “Core” Amazon EC2 and S3 security feature sets |  | 2, 4, 12 |
| Incorporating common conventional security products (Firewall, VPN) |  | 4, 12 |
| Design patterns |  | 7, 13 |
| DDoS mitigation |  | 12 |
| Encryption solutions (e.g., key services) |  | 2, 11, 12 |
| Complex access controls (building sophisticated security groups, ACLs, etc.) |  | 2, 12 |
| Amazon CloudWatch for the security architect |  | 5 |
| Trusted Advisor |  | 11 |
| CloudWatch Logs |  | 5 |
| 3.2 Recognize critical disaster recovery techniques and their implementation. |  | 3, 7, 9,  10 |
| Content may include the following: |  |  |
| Disaster recovery |  | 3 |
| Recovery time objective |  | 7 |
| Recovery point objective |  | 7 |
| Amazon Elastic Block Store |  | 3 |
| AWS Import/Export |  | 11 |
| AWS Storage Gateway |  | 11 |
| Amazon Route53 |  | 9 |
| Validation of data recovery method |  | 3 |
| 4 Domain 4.0: Troubleshooting | 10% |  |
| Content may include the following: |  |  |
| General troubleshooting information and questions |  | 5, 8 |

**Assessment Test**

1. Under a single AWS account, you have set up an Auto Scaling group with a maximum capacity of 50 Amazon Elastic Compute Cloud (Amazon EC2) instances in us-west-2. When you scale out, however, it only increases to 20 Amazon EC2 instances. What is the likely cause?

A. Auto Scaling has a hard limit of 20 Amazon EC2 instances.

B. If not specified, the Auto Scaling group maximum capacity defaults to 20 Amazon EC2 instances.

C. The Auto Scaling group desired capacity is set to 20, so Auto Scaling stopped at 20 Amazon EC2 instances.

D. You have exceeded the default Amazon EC2 instance limit of 20 per region. 2. Elastic Load Balancing allows you to distribute traffic across which of the following? A. Only within a single Availability Zone

B. Multiple Availability Zones within a region

C. Multiple Availability Zones within and between regions

D. Multiple Availability Zones within and between regions and on-premises virtualized instances running OpenStack

3. Amazon CloudWatch offers which types of monitoring plans? (Choose 2 answers) A. Basic

B. Detailed

C. Diagnostic

D. Precognitive

E. Retroactive

4. An Amazon Elastic Compute Cloud (Amazon EC2) instance in an Amazon Virtual Private Cloud (Amazon VPC) subnet can send and receive traffic from the Internet when which of the following conditions are met? (Choose 3 answers)

A. Network Access Control Lists (ACLs) and security group rules disallow all traffic except relevant Internet traffic.

B. Network ACLs and security group rules allow relevant Internet traffic.

C. Attach an Internet Gateway (IGW) to the Amazon VPC and create a subnet route table to send all non-local traffic to that IGW.

D. Attach a Virtual Private Gateway (VPG) to the Amazon VPC and create subnet routes to send all non-local traffic to that VPG.

E. The Amazon EC2 instance has a public IP address or Elastic IP (EIP) address. F. The Amazon EC2 instance does not need a public IP or Elastic IP when using

Amazon VPC.

5. If you launch five Amazon Elastic Compute Cloud (Amazon EC2) instances in an Amazon Virtual Private Cloud (Amazon VPC) without specifying a security group, the instances will be launched into a default security group that provides which of the following? (Choose 3 answers)

A. The five Amazon EC2 instances can communicate with each other. B. The five Amazon EC2 instances cannot communicate with each other. C. All inbound traffic will be allowed to the five Amazon EC2 instances. D. No inbound traffic will be allowed to the five Amazon EC2 instances. E. All outbound traffic will be allowed from the five Amazon EC2 instances. F. No outbound traffic will be allowed from the five Amazon EC2 instances.

6. Your company wants to host its secure web application in AWS. The internal security policies consider any connections to or from the web server as insecure and require application data protection. What approaches should you use to protect data in transit for the application? (Choose 2 answers)

A. Use BitLocker to encrypt data.

B. Use HTTPS with server certificate authentication.

C. Use an AWS Identity and Access Management (IAM) role.

D. Use Secure Sockets Layer (SSL)/Transport Layer Security (TLS) for database connection.

E. Use XML for data transfer from client to server.

7. You have an application that will run on an Amazon Elastic Compute Cloud (Amazon EC2) instance. The application will make requests to Amazon Simple Storage Service (Amazon S3) and Amazon DynamoDB. Using best practices, what type of AWS Identity and Access Management (IAM) identity should you create for your application to access the identified services?

A. IAM role

B. IAM user

C. IAM group

D. IAM directory

8. When a request is made to an AWS Cloud service, the request is evaluated to decide whether it should be allowed or denied. The evaluation logic follows which of the following rules? (Choose 3 answers)

A. An explicit allow overrides any denies.

B. By default, all requests are denied.

C. An explicit allow overrides the default.

D. An explicit deny overrides any allows.

E. By default, all requests are allowed.

9. What is the data processing engine behind Amazon Elastic MapReduce (Amazon EMR)? A. Apache Hadoop

B. Apache Hive

C. Apache Pig

D. Apache HBase

10. What type of AWS Elastic Beanstalk environment tier provisions resources to support a web application that handles background processing tasks?

A. Web server environment tier

B. Worker environment tier

C. Database environment tier

D. Batch environment tier

11. What Amazon Relational Database Service (Amazon RDS) feature provides the high availability for your database?

A. Regular maintenance windows

B. Security groups

C. Automated backups

D. Multi-AZ deployment

12. What administrative tasks are handled by AWS for Amazon Relational Database Service (Amazon RDS) databases? (Choose 3 answers)

A. Regular backups of the database

B. Deploying virtual infrastructure

C. Deploying the schema (for example, tables and stored procedures)

D. Patching the operating system and database software

E. Setting up non-admin database accounts and privileges

13. Which of the following use cases is well suited for Amazon Redshift? A. A 500TB data warehouse used for market analytics

B. A NoSQL, unstructured database workload

C. A high traffic, e-commerce web application

D. An in-memory cache

14. Which of the following statements about Amazon DynamoDB secondary indexes is true? A. There can be many per table, and they can be created at any time.

B. There can only be one per table, and it must be created when the table is created. C. There can be many per table, and they can be created at any time.

D. There can only be one per table, and it must be created when the table is created. 15. What is the primary use case of Amazon Kinesis Firehose?

A. Ingest huge streams of data and allow custom processing of data in flight.

B. Ingest huge streams of data and store it to Amazon Simple Storage Service (Amazon S3), Amazon Redshift, or Amazon Elasticsearch Service.

C. Generate a huge stream of data from an Amazon S3 bucket.

D. Generate a huge stream of data from Amazon DynamoDB.

16. Your company has 17TB of financial trading records that need to be stored for seven years by law. Experience has shown that any record more than a year old is unlikely to be accessed. Which of the following storage plans meets these needs in the most cost efficient manner?

A. Store the data on Amazon Elastic Block Store (Amazon EBS) volume attached to t2.large instances.

B. Store the data on Amazon Simple Storage Service (Amazon S3) with lifecycle policies that change the storage class to Amazon Glacier after one year, and delete the object after seven years.

C. Store the data in Amazon DynamoDB, and delete data older than seven years. D. Store the data in an Amazon Glacier Vault Lock.

17. What must you do to create a record of who accessed your Amazon Simple Storage Service (Amazon S3) data and from where?

A. Enable Amazon CloudWatch logs.

B. Enable versioning on the bucket.

C. Enable website hosting on the bucket.

D. Enable server access logs on the bucket.

E. Create an AWS Identity and Access Management (IAM) bucket policy.

18. Amazon Simple Storage Service (Amazon S3) is an eventually consistent storage system. For what kinds of operations is it possible to get stale data as a result of eventual consistency?

A. GET after PUT of a new object

B. GET or LIST after a DELETE

C. GET after overwrite PUT (PUT to an existing key)

D. DELETE after GET of new object

19. How is data stored in Amazon Simple Storage Service (Amazon S3) for high durability? A. Data is automatically replicated to other regions.

B. Data is automatically replicated to different Availability Zones within a region. C. Data is replicated only if versioning is enabled on the bucket.

D. Data is automatically backed up on tape and restored if needed.

20. Your company needs to provide streaming access to videos to authenticated users around the world. What is a good way to accomplish this?

A. Use Amazon Simple Storage Service (Amazon S3) buckets in each region with website hosting enabled.

B. Store the videos on Amazon Elastic Block Store (Amazon EBS) volumes. C. Enable Amazon CloudFront with geolocation and signed URLs.

D. Run a fleet of Amazon Elastic Compute Cloud (Amazon EC2) instances to host the videos.

21. Which of the following are true about the AWS shared responsibility model? (Choose 3 answers)

A. AWS is responsible for all infrastructure components (that is, AWS Cloud services) that support customer deployments.

B. The customer is responsible for the components from the guest operating system upward (including updates, security patches, and antivirus software).

C. The customer may rely on AWS to manage the security of their workloads deployed on AWS.

D. While AWS manages security of the cloud, security in the cloud is the responsibility of the customer.

E. The customer must audit the AWS data centers personally to confirm the compliance of AWS systems and services.

22. Which process in an Amazon Simple Workflow Service (Amazon SWF) workflow implements a task?

A. Decider

B. Activity worker

C. Workflow starter

D. Business rule

23. Which of the following is true if you stop an Amazon Elastic Compute Cloud (Amazon EC2) instance with an Elastic IP address in an Amazon Virtual Private Cloud (Amazon VPC)?

A. The instance is disassociated from its Elastic IP address and must be re-attached when the instance is restarted.

B. The instance remains associated with its Elastic IP address.

C. The Elastic IP address is released from your account.

D. The instance is disassociated from the Elastic IP address temporarily while you restart the instance.

24. Which Amazon Elastic Compute Cloud (Amazon EC2) pricing model allows you to pay a

set hourly price for compute, giving you full control over when the instance launches and terminates?

A. Spot instances

B. Reserved instance

C. On Demand instances

D. Dedicated instances

25. Under what circumstances will Amazon Elastic Compute Cloud (Amazon EC2) instance store data not be preserved?

A. The associated security groups are changed.

B. The instance is stopped or rebooted.

C. The instance is rebooted or terminated.

D. The instance is stopped or terminated.

E. None of the above

**Answers to Assessment Test**

1. D. Auto Scaling may cause you to reach limits of other services, such as the default number of Amazon EC2 instances you can currently launch within a region, which is 20.

2. B. The Elastic Load Balancing service allows you to distribute traffic across a group of Amazon Elastic Compute Cloud (Amazon EC2) instances in one or more Availability Zones within a region.

3. A and B. Amazon CloudWatch has two plans: basic and detailed. There are no diagnostic, precognitive, or retroactive monitoring plans for Amazon CloudWatch.

4. B, C, and E. You must do the following to create a public subnet with Internet access: Attach an IGW to your Amazon VPC.

Create a subnet route table rule to send all non-local traffic (for example, 0.0.0.0/0) to the IGW.

Configure your network ACLs and security group rules to allow relevant traffic to flow to and from your instance.

You must do the following to enable an Amazon EC2 instance to send and receive traffic from the Internet:

Assign a public IP address or EIP address.

5. A, D, and E. If a security group is not specified at launch, then an Amazon EC2 instance will be launched into the default security group for the Amazon VPC. The default security group allows communication between all resources within the security group, allows all outbound traffic, and denies all other traffic.

6. B and D. To protect data in transit from the clients to the web application, HTTPS with server certificate authentication should be used. To protect data in transit from the web application to the database, SSL/TLS for database connection should be used.

7. A. Don't create an IAM user (or an IAM group) and pass the user's credentials to the application or embed the credentials in the application. Instead, create an IAM role that you attach to the Amazon EC2 instance to give applications running on the instance temporary security credentials. The credentials have the permissions specified in the policies attached to the role. A directory is not an identity object in IAM.

8. B, C, and D. When a request is made, the AWS service decides whether a given request should be allowed or denied. The evaluation logic follows these rules:

1) By default, all requests are denied (in general, requests made using the account credentials for resources in the account are always allowed).

2) An explicit allow overrides this default.

3) An explicit deny overrides any allows.

9. A. Amazon EMR uses Apache Hadoop as its distributed data processing engine. Hadoop is an open source, Java software framework that supports data-intensive distributed

applications running on large clusters of commodity hardware. Hive, Pig, and HBase are packages that run on top of Hadoop.

10. B. An environment tier whose web application runs background jobs is known as a worker tier. An environment tier whose web application processes web requests is known as a web server tier. Database and batch are not valid environment tiers.

11. D. Multi-AZ deployment uses synchronous replication to a different Availability Zone so that operations can continue on the replica if the master database stops responding for any reason. Automated backups provide disaster recovery, not high availability. Security groups, while important, have no effect on availability. Maintenance windows are actually times when the database may not be available.

12. A, B, and D. Amazon RDS will launch Amazon Elastic Compute Cloud (Amazon EC2) instances, install the database software, handle all patching, and perform regular backups. Anything within the database software (schema, user accounts, and so on) is the responsibility of the customer.

13. A. Amazon Redshift is a petabyte-scale data warehouse. It is not well suited for unstructured NoSQL data or highly dynamic transactional data. It is in no way a cache.

14. D. There can be one secondary index per table, and it must be created when the table is created.

15. B. The Amazon Kinesis family of services provides functionality to ingest large streams of data. Amazon Kinesis Firehose is specifically designed to ingest a stream and save it to any of the three storage services listed in Response B.

16. B. Amazon S3 and Amazon Glacier are the most cost-effective storage services. After a year, when the objects are unlikely to be accessed, you can save costs by transferring the objects to Amazon Glacier where the retrieval time is three to five hours.

17. D. Server access logs provide a record of any access to an object in Amazon S3.

18. C. Amazon S3 provides read-after-write consistency for PUTs to new objects (new key), but eventual consistency for GETs and DELETEs of existing objects (existing key). Response C changes the existing object so that a subsequent GET may fetch the previous and inconsistent object.

19. B. AWS will never transfer data between regions unless directed to by you. Durability in Amazon S3 is achieved by replicating your data geographically to different Availability Zones regardless of the versioning configuration. AWS doesn't use tapes.

20. C. Amazon CloudFront provides the best user experience by delivering the data from a geographically advantageous edge location. Signed URLs allow you to control access to authenticated users.

21. A, B, and D. In the AWS shared responsibility model, customers retain control of what security they choose to implement to protect their own content, platform, applications, systems, and networks, no differently than they would for applications in an on-site data center.

22. B. An activity worker is a process or thread that performs the activity tasks that are part of your workflow. Each activity worker polls Amazon SWF for new tasks that are

appropriate for that activity worker to perform; certain tasks can be performed only by certain activity workers. After receiving a task, the activity worker processes the task to completion and then reports to Amazon SWF that the task was completed and provides

the result. The activity task represents one of the tasks that you identified in your application.

23. B. In an Amazon VPC, an instance's Elastic IP address remains associated with an instance when the instance is stopped.

24. C. You pay a set hourly price for an On Demand instance from when you launch it until you explicitly stop or terminate it. Spot instances can be terminated when the spot price goes above your bid price. Reserved instances involve paying for an instance over a one or three-year term. Dedicated instances run on hardware dedicated to your account and are not a pricing model.

25. D. The data in an instance store persists only during the lifetime of its associated instance. If an instance is stopped or terminated, then the instance store does not persist. Rebooting an instance does not shut down the instance; if an instance reboots (intentionally or unintentionally), data on the instance store persists. Security groups have nothing to do with the lifetime of an instance and have no effect here.

**Chapter 1**

**Introduction to AWS**

**THE AWS CERTIFIED SOLUTIONS ARCHITECT ASSOCIATE EXAM OBJECTIVES COVERED IN THIS CHAPTER MAY INCLUDE, BUT ARE NOT LIMITED TO, THE FOLLOWING:**

**Domain 1.0: Designing highly available, cost-efficient, fault-tolerant, scalable systems**

**1.1 Identify and recognize cloud architecture considerations, such as fundamental components and effective designs.**

**Content may include the following:**

How to design cloud services

Planning and design

Familiarity with:

Best practices for AWS architecture

Hybrid IT architectures (e.g., AWS Direct Connect, AWS Storage Gateway, Amazon Virtual Private Cloud [Amazon VPC], AWS Directory Service)

Elasticity and scalability (e.g., Auto Scaling, Amazon Simple Queue Service [Amazon SQS], Elastic Load Balancing, Amazon CloudFront)

**Domain 2.0: Implementation/Deployment**

**2.1 Identify the appropriate techniques and methods using Amazon Elastic Compute Cloud (Amazon EC2), Amazon Simple Storage Service (Amazon S3), AWS Elastic Beanstalk, AWS CloudFormation, AWS OpsWorks, Amazon VPC, and AWS Identity and Access Management (IAM) to code and implement a cloud solution.**

**Content may include the following:**

Operate and extend service management in a hybrid IT architecture.

Configure services to support compliance requirements in the cloud.

Launch instances across the AWS global infrastructure.

In 2006, Amazon Web Services, Inc. (AWS) began offering IT infrastructure services to businesses in the form of web services, now commonly known as *cloud*

*computing*. One of the key benefits of cloud computing is the opportunity to replace up-front capital infrastructure expenses with low variable costs that scale with your business. With the cloud, businesses no longer need to plan for and procure servers and other IT infrastructure weeks or months in advance. Instead, they can instantly spin up hundreds or thousands of servers in minutes and deliver results faster.

Today, AWS provides a highly reliable, scalable, and low-cost infrastructure platform in the cloud that powers hundreds of thousands of businesses in more than 190 countries around the world.

This chapter provides an introduction to the AWS Cloud computing platform. It discusses the advantages of cloud computing and the fundamentals of AWS. It provides an overview of the AWS Cloud services that are fundamentally important for the exam.

**What Is Cloud Computing?**

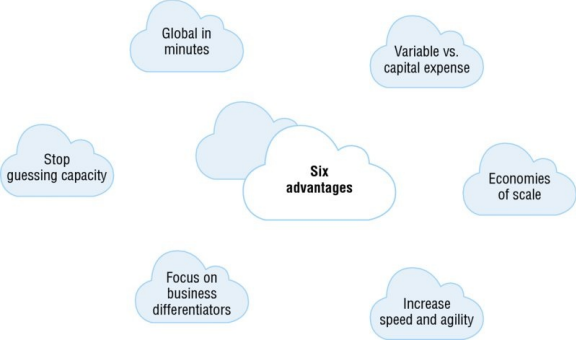
*Cloud computing* is the on-demand delivery of IT resources and applications via the Internet with pay-as-you-go pricing. Whether you run applications that share photos to millions of mobile users or deliver services that support the critical operations of your business, the cloud provides rapid access to flexible and low-cost IT resources. With cloud computing, you don’t need to make large up-front investments in hardware and spend a lot of time managing that hardware. Instead, you can provision exactly the right type and size of computing resources you need to power your newest bright idea or operate your IT department. With cloud computing, you can access as many resources as you need, almost instantly, and only pay for what you use.

In its simplest form, cloud computing provides an easy way to access servers, storage, databases, and a broad set of application services over the Internet. Cloud computing providers such as AWS own and maintain the network-connected hardware required for these application services, while you provision and use what you need for your workloads.

**Advantages of Cloud Computing**

Cloud computing introduces a revolutionary shift in how technology is obtained, used, and managed, and in how organizations budget and pay for technology services. With the ability to reconfigure the computing environment quickly to adapt to changing business requirements, organizations can optimize spending. Capacity can be automatically scaled up or down to meet fluctuating usage patterns. Services can be temporarily taken offline or shut down permanently as business demands dictate. In addition, with pay-per-use billing, AWS Cloud services become an operational expense instead of a capital expense.

While each organization experiences a unique journey to the cloud with numerous benefits, six advantages become apparent time and time again, as illustrated in Figure 1.1.

**FIGURE 1.1** Six advantages of cloud computing

**Variable vs. Capital Expense**

Let’s begin with the ability to *trade capital expense for variable operational expense*. Instead of having to invest heavily in data centers and servers before knowing how you’re going to use them, you can pay only when you consume computing resources and pay only for how much you consume.

**Economies of Scale**

Another advantage of cloud computing is that *organizations benefit from massive economies of scale*. By using cloud computing, you can achieve a lower variable cost than you would get on your own. Because usage from hundreds of thousands of customers is aggregated in the cloud, providers such as AWS can achieve higher economies of scale, which translates into lower prices.

**Stop Guessing Capacity**

When you make a capacity decision prior to deploying an application, you often end up either sitting on expensive idle resources or dealing with limited capacity. With cloud computing, organizations can *stop guessing about capacity requirements* for the infrastructure necessary to meet their business needs. They can access as much or as little as they need and scale up or down as required with only a few minutes’ notice.

**Increase Speed and Agility**

In a cloud computing environment, new IT resources are one click away, which allows

organizations to reduce the time it takes to make those resources available to developers from weeks to just minutes. This results in a dramatic *increase in speed and agility* for the organization, because the cost and time it takes to experiment and develop is significantly lower.

**Focus on Business Differentiators**

Cloud computing allows organizations to focus on their business priorities, instead of on the heavy lifting of racking, stacking, and powering servers. By embracing this paradigm shift, organizations can *stop spending money on running and maintaining data centers*. This allows organizations to focus on projects that differentiate their businesses, such as analyzing petabytes of data, delivering video content, building great mobile applications, or even exploring Mars.

**Go Global in Minutes**

Another advantage of cloud computing is the ability to *go global in minutes*. Organizations can easily deploy their applications to multiple locations around the world with just a few clicks. This allows organizations to provide redundancy across the globe and to deliver lower latency and better experiences to their customers at minimal cost. Going global used to be something only the largest enterprises could afford to do, but cloud computing democratizes this ability, making it possible for any organization.

While specific questions on these advantages of cloud computing are unlikely to be on the exam, having exposure to these benefits can help rationalize the appropriate answers.

**Cloud Computing Deployment Models**

The two primary cloud computing deployment models that the exam focuses on are “all-in” cloud-based deployments and hybrid deployments. It is important to understand how each strategy applies to architectural options and decisions.

An *all-in cloud-based application* is fully deployed in the cloud, with all components of the application running in the cloud. Applications in the cloud have either been created in the cloud or have been migrated from an existing infrastructure to take advantage of the benefits of cloud computing. Cloud-based applications can be built on low-level infrastructure pieces or can use higher-level services that provide abstraction from the management, architecting, and scaling requirements of core infrastructure.

A *hybrid deployment* is a common approach taken by many enterprises that connects infrastructure and applications between cloud-based resources and existing resources, typically in an existing data center. The most common method of hybrid deployment is between the cloud and existing on-premises infrastructure to extend and grow an organization’s infrastructure while connecting cloud resources to internal systems. Choosing between an existing investment in infrastructure and moving to the cloud does not need to be a binary decision. Leveraging dedicated connectivity, identity federation, and integrated tools allows organizations to run hybrid applications across on-premises and cloud services.

**AWS Fundamentals**

At its core, AWS provides on-demand delivery of IT resources via the Internet on a secure cloud services platform, offering compute power, storage, databases, content delivery, and other functionality to help businesses scale and grow. Using AWS resources instead of your own is like purchasing electricity from a power company instead of running your own generator, and it provides the key advantages of cloud computing: Capacity exactly matches your need, you pay only for what you use, economies of scale result in lower costs, and the service is provided by a vendor experienced in running large-scale networks.

AWS global infrastructure and AWS approach to security and compliance are key foundational concepts to understand as you prepare for the exam.

**Global Infrastructure**

AWS serves over one million active customers in more than 190 countries, and it continues to expand its global infrastructure steadily to help organizations achieve lower latency and higher throughput for their business needs.

AWS provides a highly available technology infrastructure platform with multiple locations worldwide. These locations are composed of regions and Availability Zones. Each *region* is a separate geographic area. Each region has multiple, isolated locations known as *Availability*

*Zones*. AWS enables the placement of resources and data in multiple locations. Resources aren’t replicated across regions unless organizations choose to do so.

Each region is completely independent and is designed to be completely isolated from the other regions. This achieves the greatest possible fault tolerance and stability. Each Availability Zone is also isolated, but the Availability Zones in a region are connected through low-latency links. Availability Zones are physically separated within a typical metropolitan region and are located in lower-risk flood plains (specific flood zone categorization varies by region). In addition to using a discrete uninterruptable power supply (UPS) and on-site backup generators, they are each fed via different grids from independent utilities (when available) to reduce single points of failure further. Availability Zones are all redundantly connected to multiple tier-1 transit providers. By placing resources in separate Availability Zones, you can protect your website or application from a service disruption impacting a single location.

You can achieve high availability by deploying your application across multiple Availability Zones. Redundant instances for each tier (for example, web, application, and database) of an application should be placed in distinct Availability Zones, thereby creating a multisite solution. At a minimum, the goal is to have an independent copy of each application stack in two or more Availability Zones.

**Security and Compliance**

Whether on-premises or on AWS, information security is of paramount importance to

organizations running critical workloads. Security is a core functional requirement that protects mission-critical information from accidental or deliberate theft, leakage, integrity compromise, and deletion. Helping to protect the confidentiality, integrity, and availability of systems and data is of the utmost importance to AWS, as is maintaining your trust and confidence.

This section is intended to provide a very brief introduction to AWS approach to security and compliance. Chapter 12, “Security on AWS,” and Chapter 13, “AWS Risk and Compliance,” will address these topics in greater detail, including the importance of each on the exam.

**Security**

Cloud security at AWS is the number one priority. All AWS customers benefit from data center and network architectures built to satisfy the requirements of the most security sensitive organizations. AWS and its partners offer hundreds of tools and features to help organizations meet their security objectives for visibility, auditability, controllability, and agility. This means that organizations can have the security they need, but without the capital outlay and with much lower operational overhead than in an on-premises environment.

Organizations leveraging AWS inherit all the best practices of AWS policies, architecture, and operational processes built to satisfy the requirements of the most security-sensitive customers. The AWS infrastructure has been designed to provide the highest availability while putting strong safeguards in place regarding customer privacy and segregation. When deploying systems on the AWS Cloud computing platform, AWS helps by sharing the security responsibilities with the organization. AWS manages the underlying infrastructure, and the organization can secure anything it deploys on AWS. This affords each organization the flexibility and agility they need in security controls.

This infrastructure is built and managed not only according to security best practices and standards, but also with the unique needs of the cloud in mind. AWS uses redundant and layered controls, continuous validation and testing, and a substantial amount of automation to ensure that the underlying infrastructure is monitored and protected 24/7. AWS ensures that these controls are consistently applied in every new data center or service.

**Compliance**

When customers move their production workloads to the AWS Cloud, both parties become responsible for managing the IT environment. Customers are responsible for setting up their environment in a secure and controlled manner. Customers also need to maintain adequate governance over their entire IT control environment. By tying together governance-focused, audit-friendly service features with applicable compliance or audit standards, AWS enables customers to build on traditional compliance programs. This helps organizations establish and operate in an AWS security control environment.

Organizations retain complete control and ownership over the region in which their data is physically located, allowing them to meet regional compliance and data residency requirements.

The IT infrastructure that AWS provides to organizations is designed and managed in alignment with security best practices and a variety of IT security standards. The following is a partial list of the many certifications and standards with which AWS complies:

Service Organization Controls (SOC) 1/International Standard on Assurance Engagements (ISAE) 3402, SOC 2, and SOC 3

Federal Information Security Management Act (FISMA), Department of Defense Information Assurance Certification and Accreditation Process (DIACAP), and Federal Risk and Authorization Management Program (FedRAMP)

Payment Card Industry Data Security Standard (PCI DSS) Level 1

International Organization for Standardization (ISO) 9001, ISO 27001, and ISO 27018

AWS provides a wide range of information regarding its IT control environment to help organizations achieve regulatory commitments in the form of reports, certifications, accreditations, and other third-party attestations.

**AWS Cloud Computing Platform**

AWS provides many cloud services that you can combine to meet business or organizational needs (see Figure 1.2). While being knowledgeable about all the platform services will allow you to be a well-rounded solutions architect, understanding the services and fundamental concepts outlined in this book will help prepare you for the AWS Certified Solutions Architect – Associate exam.

**FIGURE 1.2** AWS Cloud computing platform

This section introduces the major AWS Cloud services by category. Subsequent chapters provide a deeper view of the services pertinent to the exam.

**Accessing the Platform**

To access AWS Cloud services, you can use the AWS Management Console, the AWS Command Line Interface (CLI), or the AWS Software Development Kits (SDKs).

The *AWS Management Console* is a web application for managing AWS Cloud services. The console provides an intuitive user interface for performing many tasks. Each service has its own console, which can be accessed from the AWS Management Console. The console also provides information about the account and billing.

The *AWS Command Line Interface (CLI)* is a unified tool used to manage AWS Cloud services. With just one tool to download and configure, you can control multiple services from the command line and automate them through scripts.

The *AWS Software Development Kits (SDKs)* provide an application programming interface (API) that interacts with the web services that fundamentally make up the AWS platform. The SDKs provide support for many different programming languages and platforms to allow you to work with your preferred language. While you can certainly make HTTP calls directly

to the web service endpoints, using the SDKs can take the complexity out of coding by providing programmatic access for many of the services.

**Compute and Networking Services**

AWS provides a variety of compute and networking services to deliver core functionality for businesses to develop and run their workloads. These compute and networking services can be leveraged with the storage, database, and application services to provide a complete solution for computing, query processing, and storage across a wide range of applications. This section offers a high-level description of the core computing and networking services.

**Amazon Elastic Compute Cloud (Amazon EC2)**

*Amazon Elastic Compute Cloud (Amazon EC2)* is a web service that provides resizable compute capacity in the cloud. It allows organizations to obtain and configure virtual servers in Amazon’s data centers and to harness those resources to build and host software systems. Organizations can select from a variety of operating systems and resource configurations (memory, CPU, storage, and so on) that are optimal for the application profile of each workload. Amazon EC2 presents a true virtual computing environment, allowing organizations to launch compute resources with a variety of operating systems, load them with custom applications, and manage network access permissions while maintaining complete control.

**AWS Lambda**

*AWS Lambda* is a zero-administration compute platform for back-end web developers that runs your code for you on the AWS Cloud and provides you with a fine-grained pricing structure. AWS Lambda runs your back-end code on its own AWS compute fleet of Amazon EC2 instances across multiple Availability Zones in a region, which provides the high availability, security, performance, and scalability of the AWS infrastructure.

**Auto Scaling**

*Auto Scaling* allows organizations to scale Amazon EC2 capacity up or down automatically according to conditions defined for the particular workload (see Figure 1.3). Not only can it be used to help maintain application availability and ensure that the desired number of Amazon EC2 instances are running, but it also allows resources to scale in and out to match the demands of dynamic workloads. Instead of provisioning for peak load, organizations can optimize costs and use only the capacity that is actually needed.

**FIGURE 1.3** Auto scaling capacity

Auto Scaling is well suited both to applications that have stable demand patterns and to applications that experience hourly, daily, or weekly variability in usage.

**Elastic Load Balancing**

*Elastic Load Balancing* automatically distributes incoming application traffic across multiple Amazon EC2 instances in the cloud. It enables organizations to achieve greater levels of fault tolerance in their applications, seamlessly providing the required amount of load balancing capacity needed to distribute application traffic.

**AWS Elastic Beanstalk**

*AWS Elastic Beanstalk* is the fastest and simplest way to get a web application up and running on AWS. Developers can simply upload their application code, and the service automatically handles all the details, such as resource provisioning, load balancing, Auto Scaling, and monitoring. It provides support for a variety of platforms, including PHP, Java, Python, Ruby, Node.js, .NET, and Go. With AWS Elastic Beanstalk, organizations retain full control over the AWS resources powering the application and can access the underlying resources at any time.

**Amazon Virtual Private Cloud (Amazon VPC)**

*Amazon Virtual Private Cloud (Amazon VPC)* lets organizations provision a logically isolated section of the AWS Cloud where they can launch AWS resources in a virtual network that they define. Organizations have complete control over the virtual environment, including selection of the IP address range, creation of subnets, and configuration of route tables and

network gateways. In addition, organizations can extend their corporate data center networks to AWS by using hardware or software *virtual private network (VPN)* connections or dedicated circuits by using AWS Direct Connect.

**AWS Direct Connect**

*AWS Direct Connect* allows organizations to establish a dedicated network connection from their data center to AWS. Using AWS Direct Connect, organizations can establish private connectivity between AWS and their data center, office, or colocation environment, which in many cases can reduce network costs, increase bandwidth throughput, and provide a more consistent network experience than Internet-based VPN connections.

**Amazon Route 53**

*Amazon Route 53* is a highly available and scalable Domain Name System (DNS) web service. It is designed to give developers and businesses an extremely reliable and cost-effective way to route end users to Internet applications by translating human readable names, such as www.example.com, into the numeric IP addresses, such as 192.0.2.1, that computers use to connect to each other. Amazon Route 53 also serves as domain registrar, allowing you to purchase and manage domains directly from AWS.

**Storage and Content Delivery**

AWS provides a variety of services to meet your storage needs, such as Amazon Simple Storage Service, Amazon CloudFront, and Amazon Elastic Block Store. This section provides an overview of the storage and content delivery services.

**Amazon Simple Storage Service (Amazon S3)**

*Amazon Simple Storage Service (Amazon S3)* provides developers and IT teams with highly durable and scalable object storage that handles virtually unlimited amounts of data and large numbers of concurrent users. Organizations can store any number of objects of any type, such as HTML pages, source code files, image files, and encrypted data, and access them using HTTP-based protocols. Amazon S3 provides cost-effective object storage for a wide variety of use cases, including backup and recovery, nearline archive, big data analytics, disaster recovery, cloud applications, and content distribution.

**Amazon Glacier**

*Amazon Glacier* is a secure, durable, and extremely low-cost storage service for data archiving and long-term backup. Organizations can reliably store large or small amounts of data for a very low cost per gigabyte per month. To keep costs low for customers, Amazon Glacier is optimized for infrequently accessed data where a retrieval time of several hours is suitable. Amazon S3 integrates closely with Amazon Glacier to allow organizations to choose the right storage tier for their workloads.

**Amazon Elastic Block Store (Amazon EBS)**

*Amazon Elastic Block Store (Amazon EBS)* provides persistent block-level storage volumes for use with Amazon EC2 instances. Each Amazon EBS volume is automatically replicated within its Availability Zone to protect organizations from component failure, offering high

availability and durability. By delivering consistent and low-latency performance, Amazon EBS provides the disk storage needed to run a wide variety of workloads.

**AWS Storage Gateway**

*AWS Storage Gateway* is a service connecting an on-premises software appliance with cloud based storage to provide seamless and secure integration between an organization’s on premises IT environment and the AWS storage infrastructure. The service supports industry standard storage protocols that work with existing applications. It provides low-latency performance by maintaining a cache of frequently accessed data on-premises while securely storing all of your data encrypted in Amazon S3 or Amazon Glacier.

**Amazon CloudFront**

*Amazon CloudFront* is a content delivery web service. It integrates with other AWS Cloud services to give developers and businesses an easy way to distribute content to users across the world with low latency, high data transfer speeds, and no minimum usage commitments. Amazon CloudFront can be used to deliver your entire website, including dynamic, static, streaming, and interactive content, using a global network of edge locations. Requests for content are automatically routed to the nearest edge location, so content is delivered with the best possible performance to end users around the globe.

**Database Services**

AWS provides fully managed relational and NoSQL database services, and in-memory caching as a service and a petabyte-scale data warehouse solution. This section provides an overview of the products that the database services comprise.

**Amazon Relational Database Service (Amazon RDS)**

*Amazon Relational Database Service (Amazon RDS)* provides a fully managed relational database with support for many popular open source and commercial database engines. It’s a cost-efficient service that allows organizations to launch secure, highly available, fault tolerant, production-ready databases in minutes. Because Amazon RDS manages time consuming administration tasks, including backups, software patching, monitoring, scaling, and replication, organizational resources can focus on revenue-generating applications and business instead of mundane operational tasks.

**Amazon 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 fully managed database and supports both document and key/value data models. Its flexible data model and reliable performance make it a great fit for mobile, web, gaming, ad-tech, Internet of Things, and many other applications.

**Amazon Redshift**

*Amazon Redshift* is a fast, fully managed, petabyte-scale data warehouse service that makes it simple and cost effective to analyze structured data. Amazon Redshift provides a standard SQL interface that lets organizations use existing business intelligence tools. By leveraging

columnar storage technology that improves I/O efficiency and parallelizing queries across multiple nodes, Amazon Redshift is able to deliver fast query performance. The Amazon Redshift architecture allows organizations to automate most of the common administrative tasks associated with provisioning, configuring, and monitoring a cloud data warehouse.

**Amazon ElastiCache**

*Amazon ElastiCache* is a web service that simplifies deployment, operation, and scaling of an in-memory cache in the cloud. The service improves the performance of web applications by allowing organizations to retrieve information from fast, managed, in-memory caches, instead of relying entirely on slower, disk-based databases. As of this writing, Amazon ElastiCache supports Memcached and Redis cache engines.

**Management Tools**

AWS provides a variety of tools that help organizations manage your AWS resources. This section provides an overview of the management tools that AWS provides to organizations.

**Amazon CloudWatch**

*Amazon CloudWatch* is a monitoring service for AWS Cloud resources and the applications running on AWS. It allows organizations to collect and track metrics, collect and monitor log files, and set alarms. By leveraging Amazon CloudWatch, organizations can gain system-wide visibility into resource utilization, application performance, and operational health. By using these insights, organizations can react, as necessary, to keep applications running smoothly.

**AWS CloudFormation**

*AWS CloudFormation* gives developers and systems administrators an effective way to create and manage a collection of related AWS resources, provisioning and updating them in an orderly and predictable fashion. AWS CloudFormation defines a JSON-based templating language that can be used to describe all the AWS resources that are necessary for a workload. Templates can be submitted to AWS CloudFormation and the service will take care of provisioning and configuring those resources in appropriate order (see Figure 1.4).

**FIGURE 1.4** AWS CloudFormation workflow summary

**AWS CloudTrail**

*AWS CloudTrail* is a web service that records AWS API calls for an account and delivers log files for audit and review. The recorded information includes 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 service.

**AWS Config**

*AWS Config* is a fully managed service that provides organizations with an AWS resource inventory, configuration history, and configuration change notifications to enable security and governance. With AWS Config, organizations can discover existing AWS resources, export an inventory of their AWS resources with all configuration details, and determine how a resource was configured at any point in time. These capabilities enable compliance auditing, security analysis, resource change tracking, and troubleshooting.

**Security and Identity**

AWS provides security and identity services that help organizations secure their data and systems on the cloud. The following section explores these services at a high level.

**AWS Identity and Access Management (IAM)**

*AWS Identity and Access Management (IAM)* enables organizations to securely control access to AWS Cloud services and resources for their users. Using IAM, organizations can create and manage AWS users and groups and use permissions to allow and deny their access to AWS resources.

**AWS Key Management Service (KMS)**

*AWS Key Management Service (KMS)* is a managed service that makes it easy for organizations to create and control the encryption keys used to encrypt their data and uses Hardware Security Modules (HSMs) to protect the security of your keys. AWS KMS is integrated with several other AWS Cloud services to help protect data stored with these services.

**AWS Directory Service**

*AWS Directory Service* allows organizations to set up and run Microsoft Active Directory on the AWS Cloud or connect their AWS resources with an existing on-premises Microsoft Active Directory. Organizations can use it to manage users and groups, provide single sign-on to applications and services, create and apply Group Policies, domain join Amazon EC2 instances, and simplify the deployment and management of cloud-based Linux and Microsoft Windows workloads.

**AWS Certificate Manager**

*AWS Certificate Manager* is a service that lets organizations easily provision, manage, and deploy Secure Sockets Layer/Transport Layer Security (SSL/TLS) certificates for use with AWS Cloud services. It removes the time-consuming manual process of purchasing, uploading, and renewing SSL/TLS certificates. With AWS Certificate Manager, organizations

can quickly request a certificate, deploy it on AWS resources such as Elastic Load Balancing or Amazon CloudFront distributions, and let AWS Certificate Manager handle certificate renewals.

**AWS Web Application Firewall (WAF)**

*AWS Web Application Firewall (WAF)* helps protect web applications from common attacks and exploits that could affect application availability, compromise security, or consume excessive resources. AWS WAF gives organizations control over which traffic to allow or block to their web applications by defining customizable web security rules.

**Application Services**

AWS provides a variety of managed services to use with applications. The following section explores the application services at a high level.

**Amazon API Gateway**

*Amazon API Gateway* is a fully managed service that makes it easy for developers to create, publish, maintain, monitor, and secure APIs at any scale. Organizations can create an API that acts as a “front door” for applications to access data, business logic, or functionality from back-end services, such as workloads running on Amazon EC2, code running on AWS Lambda, or any web application. Amazon API Gateway handles all the tasks involved in accepting and processing up to hundreds of thousands of concurrent API calls, including traffic management, authorization and access control, monitoring, and API version management.

**Amazon Elastic Transcoder**

*Amazon Elastic Transcoder* is media transcoding in the cloud. It is designed to be a highly scalable and cost-effective way for developers and businesses to convert (or transcode) media files from their source formats into versions that will play back on devices like smartphones, tablets, and PCs.

**Amazon Simple Notification Service (Amazon SNS)**

*Amazon Simple Notification Service (Amazon SNS)* is a web service that coordinates and manages the delivery or sending of messages to recipients. In Amazon SNS, there are two types of clients—publishers and subscribers—also referred to as producers and consumers. Publishers communicate asynchronously with subscribers by producing and sending a message to a topic, which is a logical access point and communication channel. Subscribers consume or receive the message or notification over one of the supported protocols when they are subscribed to the topic.

**Amazon Simple Email Service (Amazon SES)**

*Amazon Simple Email Service (Amazon SES)* is a cost-effective email service that organizations can use to send transactional email, marketing messages, or any other type of content to their customers. Amazon SES can also be used to receive messages and deliver them to an Amazon S3 bucket, call custom code via an AWS Lambda function, or publish notifications to Amazon SNS.

**Amazon Simple Workflow Service (Amazon SWF)**

*Amazon Simple Workflow Service (Amazon SWF)* helps developers build, run, and scale background jobs that have parallel or sequential steps. Amazon SWF can be thought of as a fully managed state tracker and task coordinator on the cloud. In common architectural patterns, if your application’s steps take more than 500 milliseconds to complete, it is vitally important to track the state of processing and to provide the ability to recover or retry if a task fails. Amazon SWF helps organizations achieve this reliability.

**Amazon Simple Queue Service (Amazon SQS)**

*Amazon Simple Queue Service (Amazon SQS)* is a fast, reliable, scalable, fully managed message queuing service. Amazon SQS makes it simple and cost effective to decouple the components of a cloud application. With Amazon SQS, organizations can transmit any volume of data, at any level of throughput, without losing messages or requiring other services to be always available.

**Summary**

The term “cloud computing” refers to the on-demand delivery of IT resources via the Internet with pay-as-you-go pricing. Instead of buying, owning, and maintaining data centers and servers, organizations can acquire technology such as compute power, storage, databases, and other services on an as-needed basis. With cloud computing, AWS manages and maintains the technology infrastructure in a secure environment and businesses access these resources via the Internet to develop and run their applications. Capacity can grow or shrink instantly and businesses pay only for what they use.

Cloud computing introduces a revolutionary shift in how technology is obtained, used, and managed, and how organizations budget and pay for technology services. While each organization experiences a unique journey to the cloud with numerous benefits, six advantages become apparent time and time again. Understanding these advantages allows architects to shape solutions that deliver continuous benefits to organizations.

AWS provides a highly available technology infrastructure platform with multiple locations worldwide. These locations are composed of regions and Availability Zones. This enables organizations to place resources and data in multiple locations around the globe. Helping to protect the confidentiality, integrity, and availability of systems and data is of the utmost importance to AWS, as is maintaining the trust and confidence of organizations around the world.

AWS offers a broad set of global compute, storage, database, analytics, application, and deployment services that help organizations move faster, lower IT costs, and scale applications. Having a broad understanding of these services allows solutions architects to design effective distributed applications and systems on the AWS platform.

**Exam Essentials**

**Understand the global infrastructure.** AWS provides a highly available technology infrastructure platform with multiple locations worldwide. These locations are composed of regions and Availability Zones. Each region is located in a separate geographic area and has multiple, isolated locations known as Availability Zones.

**Understand regions.** An AWS region is a physical geographic location that consists of a cluster of data centers. AWS regions enable the placement of resources and data in multiple locations around the globe. Each region is completely independent and is designed to be completely isolated from the other regions. This achieves the greatest possible fault tolerance and stability. Resources aren’t replicated across regions unless organizations choose to do so.

**Understand Availability Zones.** An Availability Zone is one or more data centers within a region that are designed to be isolated from failures in other Availability Zones. Availability Zones provide inexpensive, low-latency network connectivity to other zones in the same region. By placing resources in separate Availability Zones, organizations can protect their website or application from a service disruption impacting a single location.

**Understand the hybrid deployment model.** A hybrid deployment model is an architectural pattern providing connectivity for infrastructure and applications between cloud-based resources and existing resources that are not located in the cloud.

**Review Questions**

1. Which of the following describes a physical location around the world where AWS clusters data centers?

A. Endpoint

B. Collection

C. Fleet

D. Region

2. Each AWS region is composed of two or more locations that offer organizations the ability to operate production systems that are more highly available, fault tolerant, and scalable than would be possible using a single data center. What are these locations called?

A. Availability Zones

B. Replication areas

C. Geographic districts

D. Compute centers

3. What is the deployment term for an environment that extends an existing on-premises infrastructure into the cloud to connect cloud resources to internal systems?

A. All-in deployment

B. Hybrid deployment

C. On-premises deployment

D. Scatter deployment

4. Which AWS Cloud service allows organizations to gain system-wide visibility into resource utilization, application performance, and operational health?

A. AWS Identity and Access Management (IAM)

B. Amazon Simple Notification Service (Amazon SNS)

C. Amazon CloudWatch

D. AWS CloudFormation

5. Which of the following AWS Cloud services is a fully managed NoSQL database service? A. Amazon Simple Queue Service (Amazon SQS)

B. Amazon DynamoDB

C. Amazon ElastiCache

D. Amazon Relational Database Service (Amazon RDS)

6. Your company experiences fluctuations in traffic patterns to their e-commerce website

based on flash sales. What service can help your company dynamically match the required compute capacity to the spike in traffic during flash sales?

A. Auto Scaling

B. Amazon Glacier

C. Amazon Simple Notification Service (Amazon SNS)

D. Amazon Virtual Private Cloud (Amazon VPC)

7. Your company provides an online photo sharing service. The development team is looking for ways to deliver image files with the lowest latency to end users so the website content is delivered with the best possible performance. What service can help speed up distribution of these image files to end users around the world?

A. Amazon Elastic Compute Cloud (Amazon EC2)

B. Amazon Route 53

C. AWS Storage Gateway

D. Amazon CloudFront

8. Your company runs an Amazon Elastic Compute Cloud (Amazon EC2) instance periodically to perform a batch processing job on a large and growing filesystem. At the end of the batch job, you shut down the Amazon EC2 instance to save money but need to persist the filesystem on the Amazon EC2 instance from the previous batch runs. What AWS Cloud service can you leverage to meet these requirements?

A. Amazon Elastic Block Store (Amazon EBS)

B. Amazon DynamoDB

C. Amazon Glacier

D. AWS CloudFormation

9. What AWS Cloud service provides a logically isolated section of the AWS Cloud where organizations can launch AWS resources in a virtual network that they define?

A. Amazon Simple Workflow Service (Amazon SWF)

B. Amazon Route 53

C. Amazon Virtual Private Cloud (Amazon VPC)

D. AWS CloudFormation

10. Your company provides a mobile voting application for a popular TV show, and 5 to 25 million viewers all vote in a 15-second timespan. What mechanism can you use to decouple the voting application from your back-end services that tally the votes?

A. AWS CloudTrail

B. Amazon Simple Queue Service (Amazon SQS)

C. Amazon Redshift

D. Amazon Simple Notification Service (Amazon SNS)

**Chapter 2**

**Amazon Simple Storage Service (Amazon S3) and Amazon Glacier Storage**

**THE AWS CERTIFIED SOLUTIONS ARCHITECT ASSOCIATE EXAM OBJECTIVES COVERED IN THIS CHAPTER MAY INCLUDE, BUT ARE NOT LIMITED TO, THE FOLLOWING:**

**Domain 1.0: Designing highly available, cost-efficient, fault-tolerant, scalable systems**

**1.1 Identify and recognize cloud architecture considerations, such as fundamental components and effective designs.**

**Content may include the following:**

How to design cloud services

Planning and design

Monitoring and logging

Familiarity with:

Best practices for AWS architecture

Developing to client specifications, including pricing/cost (e.g., On Demand vs. Reserved vs. Spot; Recovery Time Objective [RTO] and Recovery Point Objective [RPO] disaster recovery design)

Architectural trade-off decisions (e.g., high availability vs. cost)

Hybrid IT architectures

Elasticity and scalability

**Domain 2.0: Implementation/Deployment**

**2.1 Identify the appropriate techniques and methods using Amazon Simple Storage Service (Amazon S3) to code and implement a cloud solution.**

**Content may include the following:**

Configure services to support compliance requirements in the cloud.

Launch instances across the AWS global infrastructure.

Configure AWS Identity and Access Management (IAM) policies and best practices. **Domain 3.0: Data Security**

**3.1 Recognize and implement secure practices for optimum cloud deployment and maintenance**

**Content may include the following:**

Security Architecture with AWS

“Core” Amazon S3 security feature sets

Encryption solutions (e.g., key services)

Complex access controls (building sophisticated security groups, Access Control Lists [ACLs], etc.)



**Introduction**

This chapter is intended to provide you with a basic understanding of the core object storage services available on AWS: Amazon Simple Storage Service (Amazon S3) and Amazon Glacier.

Amazon S3 provides developers and IT teams with secure, durable, and highly-scalable cloud storage. Amazon S3 is easy-to-use *object storage* with a simple web service interface that you can use to store and retrieve any amount of data from anywhere on the web. Amazon S3 also allows you to pay only for the storage you actually use, which eliminates the capacity

planning and capacity constraints associated with traditional storage.

Amazon S3 is one of first services introduced by AWS, and it serves as one of the foundational web services—nearly any application running in AWS uses Amazon S3, either directly or indirectly. Amazon S3 can be used alone or in conjunction with other AWS services, and it offers a very high level of integration with many other AWS cloud services. For example, Amazon S3 serves as the durable target storage for Amazon Kinesis and Amazon Elastic MapReduce (Amazon EMR), it is used as the storage for Amazon Elastic Block Store (Amazon EBS) and Amazon Relational Database Service (Amazon RDS) snapshots, and it is used as a data staging or loading storage mechanism for Amazon Redshift and Amazon DynamoDB, among many other functions. Because Amazon S3 is so flexible, so highly integrated, and so commonly used, it is important to understand this service in detail.

Common use cases for Amazon S3 storage include:

Backup and archive for on-premises or cloud data

Content, media, and software storage and distribution

Big data analytics

Static website hosting

Cloud-native mobile and Internet application hosting

Disaster recovery

To support these use cases and many more, Amazon S3 offers a range of *storage classes* designed for various generic use cases: general purpose, infrequent access, and archive. To help manage data through its lifecycle, Amazon S3 offers configurable lifecycle policies. By using lifecycle policies, you can have your data automatically migrate to the most appropriate storage class, without modifying your application code. In order to control who has access to your data, Amazon S3 provides a rich set of permissions, access controls, and encryption options.

Amazon Glacier is another cloud storage service related to Amazon S3, but optimized for data archiving and long-term backup at extremely low cost. Amazon Glacier is suitable for “cold data,” which is data that is rarely accessed and for which a retrieval time of three to five hours is acceptable. Amazon Glacier can be used both as a storage class of Amazon S3 (see Storage Classes and Object Lifecycle Management topics in the Amazon S3 Advanced Features section), and as an independent archival storage service (see the Amazon Glacier section).

**Object Storage versus Traditional Block and File Storage**

In traditional IT environments, two kinds of storage dominate: *block storage* and *file storage*. Block storage operates at a lower level—the raw storage device level—and manages data as a set of numbered, fixed-size blocks. File storage operates at a higher level—the operating system level—and manages data as a named hierarchy of files and folders. Block and file storage are often accessed over a network in the form of a Storage Area Network (SAN) for block storage, using protocols such as iSCSI or Fibre Channel, or as a Network Attached Storage (NAS) file server or “filer” for file storage, using protocols such as Common Internet File System (CIFS) or Network File System (NFS). Whether directly-attached or network attached, block or file, this kind of storage is very closely associated with the server and the operating system that is using the storage.

Amazon S3 object storage is something quite different. Amazon S3 is cloud *object storage*. Instead of being closely associated with a server, Amazon S3 storage is independent of a server and is accessed over the Internet. Instead of managing data as blocks or files using SCSI, CIFS, or NFS protocols, data is managed as objects using an Application Program Interface (API) built on standard HTTP verbs.

Each Amazon S3 object contains both data and metadata. Objects reside in containers called *buckets*, and each object is identified by a unique user-specified key (filename). Buckets are a simple flat folder with no file system hierarchy. That is, you can have multiple buckets, but you can’t have a sub-bucket within a bucket. Each bucket can hold an unlimited number of objects.

It is easy to think of an Amazon S3 object (or the data portion of an object) as a file, and the key as the filename. However, keep in mind that Amazon S3 is not a traditional file system and differs in significant ways. In Amazon S3, you GET an object or PUT an object, operating on the whole object at once, instead of incrementally updating portions of the object as you would with a file. You can’t “mount” a bucket, “open” an object, install an operating system on Amazon S3, or run a database on it.

Instead of a file system, Amazon S3 is highly-durable and highly-scalable object storage that is optimized for reads and is built with an intentionally minimalistic feature set. It provides a simple and robust abstraction for file storage that frees you from many underlying details that you normally do have to deal with in traditional storage. For example, with Amazon S3 you don’t have to worry about device or file system storage limits and capacity planning—a single bucket can store an unlimited number of files. You also don’t need to worry about data durability or replication across availability zones—Amazon S3 objects are automatically replicated on multiple devices in multiple facilities within a region. The same with scalability —if your request rate grows steadily, Amazon S3 automatically partitions buckets to support very high request rates and simultaneous access by many clients.

If you need traditional block or file storage in addition to Amazon S3 storage, AWS provides options. The Amazon EBS service provides block level storage for Amazon Elastic Compute Cloud (Amazon EC2) instances. Amazon Elastic File System (AWS EFS) provides network-attached shared file storage (NAS storage) using the NFS v4 protocol.

**Amazon Simple Storage Service (Amazon S3) Basics**

Now that you have an understanding of some of the key differences between traditional block and file storage versus cloud object storage, we can explore the basics of Amazon S3 in more detail.

**Buckets**

A *bucket* is a container (web folder) for objects (files) stored in Amazon S3. Every Amazon S3 object is contained in a bucket. Buckets form the top-level namespace for Amazon S3, and bucket names are global. This means that your bucket names must be unique across all AWS accounts, much like Domain Name System (DNS) domain names, not just within your own account. Bucket names can contain up to 63 lowercase letters, numbers, hyphens, and periods. You can create and use multiple buckets; you can have up to 100 per account by default.

It is a best practice to use bucket names that contain your domain name and conform to the rules for DNS names. This ensures that your bucket names are your own, can be used in all regions, and can host static websites.

**AWS Regions**

Even though the namespace for Amazon S3 buckets is global, each Amazon S3 bucket is created in a specific region that you choose. This lets you control where your data is stored. You can create and use buckets that are located close to a particular set of end users or customers in order to minimize latency, or located in a particular region to satisfy data locality and sovereignty concerns, or located far away from your primary facilities in order to satisfy disaster recovery and compliance needs. You control the location of your data; data in an Amazon S3 bucket is stored in that region unless you explicitly copy it to another bucket located in a different region.

**Objects**

*Objects* are the entities or files stored in Amazon S3 buckets. An object can store virtually any kind of data in any format. Objects can range in size from 0 bytes up to 5TB, and a single bucket can store an unlimited number of objects. This means that Amazon S3 can store a virtually unlimited amount of data.

Each object consists of data (the file itself) and *metadata* (data about the file). The data portion of an Amazon S3 object is opaque to Amazon S3. This means that an object’s data is treated as simply a stream of bytes—Amazon S3 doesn’t know or care what type of data you are storing, and the service doesn’t act differently for text data versus binary data.

The metadata associated with an Amazon S3 object is a set of name/value pairs that describe the object. There are two types of metadata: system metadata and user metadata. System metadata is created and used by Amazon S3 itself, and it includes things like the date last modified, object size, MD5 digest, and HTTP Content-Type. User metadata is optional, and it can only be specified at the time an object is created. You can use custom metadata to tag your data with attributes that are meaningful to you.

**Keys**

Every object stored in an S3 bucket is identified by a unique identifier called a *key*. You can think of the key as a filename. A key can be up to 1024 bytes of Unicode UTF-8 characters, including embedded slashes, backslashes, dots, and dashes.

Keys must be unique within a single bucket, but different buckets can contain objects with the same key. The combination of bucket, key, and optional version ID uniquely identifies an Amazon S3 object.

**Object URL**

Amazon S3 is storage for the Internet, and every Amazon S3 object can be addressed by a unique URL formed using the web services endpoint, the bucket name, and the object key. For example, with the URL:

http://mybucket.s3.amazonaws.com/jack.doc

mybucket is the S3 bucket name, and jack.doc is the key or filename. If another object is created, for instance:

http://mybucket.s3.amazonaws.com/fee/fi/fo/fum/jack.doc

then the bucket name is still mybucket, but now the key or filename is the string fee/fi/fo/fum/jack.doc. A key may contain delimiter characters like slashes or backslashes to help you name and logically organize your Amazon S3 objects, but to Amazon S3 it is simply a long key name in a flat namespace. There is no actual file and folder hierarchy. See the topic “Prefixes and Delimiters” in the “Amazon S3 Advanced Features” section that follows for more information.

For convenience, the Amazon S3 console and the Prefix and Delimiter feature allow you to navigate within an Amazon S3 bucket as if there were a folder hierarchy. However, remember that a bucket is a single flat namespace of keys with no structure.

**Amazon S3 Operations**

The Amazon S3 API is intentionally simple, with only a handful of common operations. They include:

Create/delete a bucket

Write an object

Read an object

Delete an object

List keys in a bucket

**REST Interface**

The native interface for Amazon S3 is a *REST (Representational State Transfer)* API. With the REST interface, you use standard HTTP or HTTPS requests to create and delete buckets, list keys, and read and write objects. REST maps standard HTTP “verbs” (HTTP methods) to

the familiar CRUD (Create, Read, Update, Delete) operations. Create is HTTP PUT (and sometimes POST); read is HTTP GET; delete is HTTP DELETE; and update is HTTP POST (or sometimes PUT).

Always use HTTPS for Amazon S3 API requests to ensure that your requests and data are secure.

In most cases, users do not use the REST interface directly, but instead interact with Amazon S3 using one of the higher-level interfaces available. These include the AWS Software Development Kits (SDKs) (wrapper libraries) for iOS, Android, JavaScript, Java, .NET, Node.js, PHP, Python, Ruby, Go, and C++, the AWS Command Line Interface (CLI), and the AWS Management Console.

Amazon S3 originally supported a SOAP (Simple Object Access Protocol) API in addition to the REST API, but you should use the REST API. The legacy HTTPS endpoint is still available, but new features are not supported.

**Durability and Availability**

Data *durability* and *availability* are related but slightly different concepts. Durability addresses the question, “Will my data still be there in the future?” Availability addresses the question, “Can I access my data right now?” Amazon S3 is designed to provide both very high durability and very high availability for your data.

Amazon S3 standard storage is designed for 99.999999999% durability and 99.99% availability of objects over a given year. For example, if you store 10,000 objects with Amazon S3, you can on average expect to incur a loss of a single object once every 10,000,000 years. Amazon S3 achieves high durability by automatically storing data redundantly on multiple devices in multiple facilities within a region. It is designed to sustain the concurrent loss of data in two facilities without loss of user data. Amazon S3 provides a highly durable storage infrastructure designed for mission-critical and primary data storage.

If you need to store non-critical or easily reproducible derived data (such as image thumbnails) that doesn’t require this high level of durability, you can choose to use Reduced Redundancy Storage (RRS) at a lower cost. RRS offers 99.99% durability with a lower cost of storage than traditional Amazon S3 storage.

Even though Amazon S3 storage offers very high durability at the infrastructure level, it is still a best practice to protect against user-level accidental deletion or overwriting of data by using additional features such as versioning, cross-region replication, and MFA Delete.

**Data Consistency**

Amazon S3 is an *eventually consistent* system. Because your data is automatically replicated across multiple servers and locations within a region, changes in your data may take some time to propagate to all locations. As a result, there are some situations where information that you read immediately after an update may return stale data.

For PUTs to new objects, this is not a concern—in this case, Amazon S3 provides read-after write consistency. However, for PUTs to existing objects (object overwrite to an existing key) and for object DELETEs, Amazon S3 provides *eventual consistency*.

Eventual consistency means that if you PUT new data to an existing key, a subsequent GET might return the old data. Similarly, if you DELETE an object, a subsequent GET for that object might still read the deleted object. In all cases, updates to a single key are atomic—for eventually-consistent reads, you will get the new data or the old data, but never an inconsistent mix of data.

**Access Control**

Amazon S3 is secure by default; when you create a bucket or object in Amazon S3, only you have access. To allow you to give controlled access to others, Amazon S3 provides both coarse-grained access controls (Amazon S3 Access Control Lists [ACLs]), and fine-grained access controls (Amazon S3 bucket policies, AWS Identity and Access Management [IAM] policies, and query-string authentication).

Amazon S3 ACLs allow you to grant certain coarse-grained permissions: READ, WRITE, or FULL-CONTROL at the object or bucket level. ACLs are a legacy access control mechanism, created before IAM existed. ACLs are best used today for a limited set of use cases, such as enabling bucket logging or making a bucket that hosts a static website be world-readable.

Amazon S3 bucket policies are the recommended access control mechanism for Amazon S3 and provide much finer-grained control. Amazon S3 bucket policies are very similar to IAM policies, which were discussed in Chapter 6, “AWS Identity and Access Management (IAM),” but are subtly different in that:

They are associated with the bucket resource instead of an IAM principal.

They include an explicit reference to the IAM principal in the policy. This principal can be associated with a different AWS account, so Amazon S3 bucket policies allow you to assign cross-account access to Amazon S3 resources.

Using an Amazon S3 bucket policy, you can specify who can access the bucket, from where (by Classless Inter-Domain Routing [CIDR] block or IP address), and during what time of day.

Finally, IAM policies may be associated directly with IAM principals that grant access to an Amazon S3 bucket, just as it can grant access to any AWS service and resource. Obviously, you can only assign IAM policies to principals in AWS accounts that you control.

**Static Website Hosting**

A very common use case for Amazon S3 storage is *static website* hosting. Many websites, particularly micro-sites, don’t need the services of a full web server. A static website means

that all of the pages of the website contain only static content and do not require server-side processing such as PHP, ASP.NET, or JSP. (Note that this does not mean that the website cannot be interactive and dynamic; this can be accomplished with client-side scripts, such as JavaScript embedded in static HTML webpages.) Static websites have many advantages: they are very fast, very scalable, and can be more secure than a typical dynamic website. If you host a static website on Amazon S3, you can also leverage the security, durability, availability, and scalability of Amazon S3.

Because every Amazon S3 object has a URL, it is relatively straightforward to turn a bucket into a website. To host a static website, you simply configure a bucket for website hosting and then upload the content of the static website to the bucket.

To configure an Amazon S3 bucket for static website hosting:

1. Create a bucket with the same name as the desired website hostname. 2. Upload the static files to the bucket.

3. Make all the files public (world readable).

4. Enable static website hosting for the bucket. This includes specifying an Index document and an Error document.

5. The website will now be available at the S3 website URL:

<bucket-name>.s3-website-<AWS-region>.amazonaws.com.

6. Create a friendly DNS name in your own domain for the website using a DNS CNAME, or an Amazon Route 53 alias that resolves to the Amazon S3 website URL.

7. The website will now be available at your website domain name.

**Amazon S3 Advanced Features**

Beyond the basics, there are some advanced features of Amazon S3 that you should also be familiar with.

**Prefixes and Delimiters**

While Amazon S3 uses a flat structure in a bucket, it supports the use of *prefix* and *delimiter* parameters when listing key names. This feature lets you organize, browse, and retrieve the objects within a bucket hierarchically. Typically, you would use a slash (/) or backslash (\) as a delimiter and then use key names with embedded delimiters to emulate a file and folder hierarchy within the flat object key namespace of a bucket.

For example, you might want to store a series of server logs by server name (such as server42), but organized by year and month, like so:

logs/2016/January/server42.log

logs/2016/February/server42.log

logs/2016/March/server42.log

The REST API, wrapper SDKs, AWS CLI, and the Amazon Management Console all support the use of delimiters and prefixes. This feature lets you logically organize new data and easily maintain the hierarchical folder-and-file structure of existing data uploaded or backed up from traditional file systems. Used together with IAM or Amazon S3 bucket policies, prefixes and delimiters also allow you to create the equivalent of departmental “subdirectories” or user “home directories” within a single bucket, restricting or sharing access to these “subdirectories” (defined by prefixes) as needed.

Use delimiters and object prefixes to hierarchically organize the objects in your Amazon S3 buckets, but always remember that Amazon S3 is not really a file system.

**Storage Classes**

Amazon S3 offers a range of *storage classes* suitable for various use cases.

*Amazon S3 Standard* offers high durability, high availability, low latency, and high performance object storage for general purpose use. Because it delivers low first-byte latency and high throughput, Standard is well-suited for short-term or long-term storage of frequently accessed data. For most general purpose use cases, Amazon S3 Standard is the place to start.

*Amazon S3 Standard – Infrequent Access (Standard-IA)* offers the same durability, low latency, and high throughput as Amazon S3 Standard, but is designed for long-lived, less frequently accessed data. Standard-IA has a lower per GB-month storage cost than Standard, but the price model also includes a minimum object size (128KB), minimum duration (30 days), and per-GB retrieval costs, so it is best suited for infrequently accessed data that is stored for longer than 30 days.

*Amazon S3 Reduced Redundancy Storage (RRS)* offers slightly lower durability (4 nines) than Standard or Standard-IA at a reduced cost. It is most appropriate for derived data that can be easily reproduced, such as image thumbnails.

Finally, the *Amazon Glacier* storage class offers secure, durable, and extremely low-cost cloud storage for data that does not require real-time access, such as archives and long-term backups. To keep costs low, Amazon Glacier is optimized for infrequently accessed data where a retrieval time of several hours is suitable. To retrieve an Amazon Glacier object, you issue a restore command using one of the Amazon S3 APIs; three to five hours later, the Amazon Glacier object is copied to Amazon S3 RRS. Note that the restore simply creates a copy in Amazon S3 RRS; the original data object remains in Amazon Glacier until explicitly deleted. Also be aware that Amazon Glacier allows you to retrieve up to 5% of the Amazon S3 data stored in Amazon Glacier for free each month; restores beyond the daily restore allowance incur a restore fee. Refer to the Amazon Glacier pricing page on the AWS website for full details.

In addition to acting as a storage tier in Amazon S3, Amazon Glacier is also a standalone storage service with a separate API and some unique characteristics. However, when you use Amazon Glacier as a storage class of Amazon S3, you always interact with the data via the Amazon S3 APIs. Refer to the Amazon Glacier section for more details.

Set a data retrieval policy to limit restores to the free tier or to a maximum GB per-hour limit to avoid or minimize Amazon Glacier restore fees.

**Object Lifecycle Management**

Amazon S3 *Object Lifecycle Management* is roughly equivalent to automated *storage tiering* in traditional IT storage infrastructures. In many cases, data has a natural lifecycle, starting out as “hot” (frequently accessed) data, moving to “warm” (less frequently accessed) data as it ages, and ending its life as “cold” (long-term backup or archive) data before eventual deletion.

For example, many business documents are frequently accessed when they are created, then become much less frequently accessed over time. In many cases, however, compliance rules require business documents to be archived and kept accessible for years. Similarly, studies show that file, operating system, and database backups are most frequently accessed in the first few days after they are created, usually to restore after an inadvertent error. After a week or two, these backups remain a critical asset, but they are much less likely to be accessed for a restore. In many cases, compliance rules require that a certain number of backups be kept for several years.

Using Amazon S3 lifecycle configuration rules, you can significantly reduce your storage costs by automatically transitioning data from one storage class to another or even automatically deleting data after a period of time. For example, the lifecycle rules for backup data might be:

Store backup data initially in Amazon S3 Standard.

After 30 days, transition to Amazon Standard-IA.

After 90 days, transition to Amazon Glacier.

After 3 years, delete.

Lifecycle configurations are attached to the bucket and can apply to all objects in the bucket or only to objects specified by a prefix.

**Encryption**

It is strongly recommended that all sensitive data stored in Amazon S3 be encrypted, both in flight and at rest.

To encrypt your Amazon S3 data in flight, you can use the Amazon S3 Secure Sockets Layer (SSL) API endpoints. This ensures that all data sent to and from Amazon S3 is encrypted while in transit using the HTTPS protocol.

To encrypt your Amazon S3 data at rest, you can use several variations of *Server-Side Encryption (SSE)*. Amazon S3 encrypts your data at the object level as it writes it to disks in its data centers and decrypts it for you when you access it. All SSE performed by Amazon S3 and AWS Key Management Service (Amazon KMS) uses the 256-bit Advanced Encryption Standard (AES). You can also encrypt your Amazon S3 data at rest using *Client-Side Encryption*, encrypting your data on the client before sending it to Amazon S3.

**SSE-S3 (AWS-Managed Keys)**

This is a fully integrated “check-box-style” encryption solution where AWS handles the key management and key protection for Amazon S3. Every object is encrypted with a unique key. The actual object key itself is then further encrypted by a separate master key. A new master key is issued at least monthly, with AWS rotating the keys. Encrypted data, encryption keys, and master keys are all stored separately on secure hosts, further enhancing protection.

**SSE-KMS (AWS KMS Keys)**

This is a fully integrated solution where Amazon handles your key management and protection for Amazon S3, but where you manage the keys. SSE-KMS offers several additional benefits compared to SSE-S3. Using SSE-KMS, there are separate permissions for using the master key, which provide protection against unauthorized access to your objects stored in Amazon S3 and an additional layer of control. AWS KMS also provides auditing, so you can see who used your key to access which object and when they tried to access this object. AWS KMS also allows you to view any failed attempts to access data from users who did not have permission to decrypt the data.

**SSE-C (Customer-Provided Keys)**

This is used when you want to maintain your own encryption keys but don’t want to manage or implement your own client-side encryption library. With SSE-C, AWS will do the encryption/decryption of your objects while you maintain full control of the keys used to encrypt/decrypt the objects in Amazon S3.

**Client-Side Encryption**

Client-side encryption refers to encrypting data on the client side of your application before sending it to Amazon S3. You have the following two options for using data encryption keys:

Use an AWS KMS-managed customer master key.

Use a client-side master key.

When using client-side encryption, you retain end-to-end control of the encryption process, including management of the encryption keys.

For maximum simplicity and ease of use, use server-side encryption with AWS managed keys (SSE-S3 or SSE-KMS).

**Versioning**

Amazon S3 versioning helps protects your data against accidental or malicious deletion by keeping multiple versions of each object in the bucket, identified by a unique version ID. Versioning allows you to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. If a user makes an accidental change or even maliciously deletes an object in your S3 bucket, you can restore the object to its original state simply by referencing the version ID in addition to the bucket and object key. Versioning is turned on at the bucket level. Once enabled, versioning cannot be removed from a bucket; it can only be suspended.

**MFA Delete**

*MFA Delete* adds another layer of data protection on top of bucket versioning. MFA Delete requires additional authentication in order to permanently delete an object version or change the versioning state of a bucket. In addition to your normal security credentials, MFA Delete requires an authentication code (a temporary, one-time password) generated by a hardware or virtual Multi-Factor Authentication (MFA) device. Note that MFA Delete can only be enabled by the root account.

**Pre-Signed URLs**

All Amazon S3 objects by default are private, meaning that only the owner has access. However, the object owner can optionally share objects with others by creating a *pre-signed URL*, using their own security credentials to grant time-limited permission to download the objects. When you create a pre-signed URL for your object, you must provide your security credentials and specify a bucket name, an object key, the HTTP method (GET to download the object), and an expiration date and time. The pre-signed URLs are valid only for the specified duration. This is particularly useful to protect against “content scraping” of web content such as media files stored in Amazon S3.

**Multipart Upload**

To better support uploading or copying of large objects, Amazon S3 provides the Multipart Upload API. This allows you to upload large objects as a set of parts, which generally gives better network utilization (through parallel transfers), the ability to pause and resume, and the ability to upload objects where the size is initially unknown.

Multipart upload is a three-step process: initiation, uploading the parts, and completion (or

abort). Parts can be uploaded independently in arbitrary order, with retransmission if needed. After all of the parts are uploaded, Amazon S3 assembles the parts in order to create an object.

In general, you *should* use multipart upload for objects larger than 100 Mbytes, and you *must* use multipart upload for objects larger than 5GB. When using the low-level APIs, you must break the file to be uploaded into parts and keep track of the parts. When using the high-level APIs and the high-level Amazon S3 commands in the AWS CLI (aws s3 cp, aws s3 mv, and aws s3 sync), multipart upload is automatically performed for large objects.

You can set an object lifecycle policy on a bucket to abort incomplete multipart uploads after a specified number of days. This will minimize the storage costs associated with multipart uploads that were not completed.

**Range GETs**

It is possible to download (GET) only a portion of an object in both Amazon S3 and Amazon Glacier by using something called a *Range GET*. Using the Range HTTP header in the GET request or equivalent parameters in one of the SDK wrapper libraries, you specify a range of bytes of the object. This can be useful in dealing with large objects when you have poor connectivity or to download only a known portion of a large Amazon Glacier backup.

**Cross-Region Replication**

*Cross-region replication* is a feature of Amazon S3 that allows you to asynchronously replicate all new objects in the source bucket in one AWS region to a target bucket in another region. Any metadata and ACLs associated with the object are also part of the replication. After you set up cross-region replication on your source bucket, any changes to the data, metadata, or ACLs on an object trigger a new replication to the destination bucket. To enable cross-region replication, versioning must be turned on for both source and destination buckets, and you must use an IAM policy to give Amazon S3 permission to replicate objects on your behalf.

Cross-region replication is commonly used to reduce the latency required to access objects in Amazon S3 by placing objects closer to a set of users or to meet requirements to store backup data at a certain distance from the original source data.

If turned on in an existing bucket, cross-region replication will only replicate new objects. Existing objects will not be replicated and must be copied to the new bucket via a separate command.

**Logging**

In order to track requests to your Amazon S3 bucket, you can enable Amazon S3 server access logs. Logging is off by default, but it can easily be enabled. When you enable logging for a

bucket (the source bucket), you must choose where the logs will be stored (the target bucket). You can store access logs in the same bucket or in a different bucket. Either way, it is optional (but a best practice) to specify a prefix, such as logs/ or yourbucketname/logs/, so that you can more easily identify your logs.

Once enabled, logs are delivered on a best-effort basis with a slight delay. Logs include information such as:

Requestor account and IP address

Bucket name

Request time

Action (GET, PUT, LIST, and so forth)

Response status or error code

**Event Notifications**

Amazon S3 *event notifications* can be sent in response to actions taken on objects uploaded or stored in Amazon S3. Event notifications enable you to run workflows, send alerts, or perform other actions in response to changes in your objects stored in Amazon S3. You can use Amazon S3 event notifications to set up triggers to perform actions, such as transcoding media files when they are uploaded, processing data files when they become available, and synchronizing Amazon S3 objects with other data stores.

Amazon S3 event notifications are set up at the bucket level, and you can configure them through the Amazon S3 console, through the REST API, or by using an AWS SDK. Amazon S3 can publish notifications when new objects are created (by a PUT, POST, COPY, or multipart upload completion), when objects are removed (by a DELETE), or when Amazon S3 detects that an RRS object was lost. You can also set up event notifications based on object name prefixes and suffixes. Notification messages can be sent through either Amazon Simple Notification Service (Amazon SNS) or Amazon Simple Queue Service (Amazon SQS) or delivered directly to AWS Lambda to invoke AWS Lambda functions.

**Best Practices, Patterns, and Performance**

It is a common pattern to use Amazon S3 storage in hybrid IT environments and applications. For example, data in on-premises file systems, databases, and compliance archives can easily be backed up over the Internet to Amazon S3 or Amazon Glacier, while the primary application or database storage remains on-premises.

Another common pattern is to use Amazon S3 as bulk “blob” storage for data, while keeping an index to that data in another service, such as Amazon DynamoDB or Amazon RDS. This allows quick searches and complex queries on key names without listing keys continually.

Amazon S3 will scale automatically to support very high request rates, automatically re partitioning your buckets as needed. If you need request rates higher than 100 requests per second, you may want to review the Amazon S3 best practices guidelines in the Developer Guide. To support higher request rates, it is best to ensure some level of random distribution of keys, for example by including a hash as a prefix to key names.

If you are using Amazon S3 in a GET-intensive mode, such as a static website hosting, for best performance you should consider using an Amazon CloudFront distribution as a caching layer in front of your Amazon S3 bucket.

**Amazon Glacier**

Amazon Glacier is an extremely low-cost storage service that provides durable, secure, and flexible storage for data archiving and online backup. To keep costs low, Amazon Glacier is designed for infrequently accessed data where a retrieval time of three to five hours is acceptable.

Amazon Glacier can store an unlimited amount of virtually any kind of data, in any format. Common use cases for Amazon Glacier include replacement of traditional tape solutions for long-term backup and archive and storage of data required for compliance purposes. In most cases, the data stored in Amazon Glacier consists of large TAR (Tape Archive) or ZIP files.

Like Amazon S3, Amazon Glacier is extremely durable, storing data on multiple devices across multiple facilities in a region. Amazon Glacier is designed for 99.999999999% durability of objects over a given year.

**Archives**

In Amazon Glacier, data is stored in *archives*. An archive can contain up to 40TB of data, and you can have an unlimited number of archives. Each archive is assigned a unique archive ID at the time of creation. (Unlike an Amazon S3 object key, you cannot specify a user-friendly archive name.) All archives are automatically encrypted, and archives are immutable—after an archive is created, it cannot be modified.

**Vaults**

*Vaults* are containers for archives. Each AWS account can have up to 1,000 vaults. You can control access to your vaults and the actions allowed using IAM policies or vault access policies.

**Vaults Locks**

You can easily deploy and enforce compliance controls for individual Amazon Glacier vaults with a *vault lock* policy. You can specify controls such as Write Once Read Many (WORM) in a vault lock policy and lock the policy from future edits. Once locked, the policy can no longer be changed.

**Data Retrieval**

You can retrieve up to 5% of your data stored in Amazon Glacier for free each month, calculated on a daily prorated basis. If you retrieve more than 5%, you will incur retrieval fees based on your maximum retrieval rate. To eliminate or minimize those fees, you can set a data retrieval policy on a vault to limit your retrievals to the free tier or to a specified data rate.

**Amazon Glacier versus Amazon Simple Storage Service (Amazon S3)**

Amazon Glacier is similar to Amazon S3, but it differs in several key aspects. Amazon Glacier supports 40TB archives versus 5TB objects in Amazon S3. Archives in Amazon Glacier are

identified by system-generated archive IDs, while Amazon S3 lets you use “friendly” key names. Amazon Glacier archives are automatically encrypted, while encryption at rest is optional in Amazon S3. However, by using Amazon Glacier as an Amazon S3 storage class together with object lifecycle policies, you can use the Amazon S3 interface to get most of the benefits of Amazon Glacier without learning a new interface.

**Summary**

Amazon S3 is the core object storage service on AWS, allowing you to store an unlimited amount of data with very high durability.

Common Amazon S3 use cases include backup and archive, web content, big data analytics, static website hosting, mobile and cloud-native application hosting, and disaster recovery.

Amazon S3 is integrated with many other AWS cloud services, including AWS IAM, AWS KMS, Amazon EC2, Amazon EBS, Amazon EMR, Amazon DynamoDB, Amazon Redshift, Amazon SQS, AWS Lambda, and Amazon CloudFront.

Object storage differs from traditional block and file storage. Block storage manages data at a device level as addressable blocks, while file storage manages data at the operating system level as files and folders. Object storage manages data as objects that contain both data and metadata, manipulated by an API.

Amazon S3 buckets are containers for objects stored in Amazon S3. Bucket names must be globally unique. Each bucket is created in a specific region, and data does not leave the region unless explicitly copied by the user.

Amazon S3 objects are files stored in buckets. Objects can be up to 5TB and can contain any kind of data. Objects contain both data and metadata and are identified by keys. Each Amazon S3 object can be addressed by a unique URL formed by the web services endpoint, the bucket name, and the object key.

Amazon S3 has a minimalistic API—create/delete a bucket, read/write/delete objects, list keys in a bucket—and uses a REST interface based on standard HTTP verbs—GET, PUT, POST, and DELETE. You can also use SDK wrapper libraries, the AWS CLI, and the AWS Management Console to work with Amazon S3.

Amazon S3 is highly durable and highly available, designed for 11 nines of durability of objects in a given year and four nines of availability.

Amazon S3 is eventually consistent, but offers read-after-write consistency for new object PUTs.

Amazon S3 objects are private by default, accessible only to the owner. Objects can be marked public readable to make them accessible on the web. Controlled access may be provided to others using ACLs and AWS IAM and Amazon S3 bucket policies.

Static websites can be hosted in an Amazon S3 bucket.

Prefixes and delimiters may be used in key names to organize and navigate data hierarchically much like a traditional file system.

Amazon S3 offers several storage classes suited to different use cases: Standard is designed for general-purpose data needing high performance and low latency. Standard-IA is for less frequently accessed data. RRS offers lower redundancy at lower cost for easily reproduced data. Amazon Glacier offers low-cost durable storage for archive and long-term backups that can are rarely accessed and can accept a three- to five-hour retrieval time.

Object lifecycle management policies can be used to automatically move data between

storage classes based on time.

Amazon S3 data can be encrypted using server-side or client-side encryption, and encryption keys can be managed with Amazon KMS.

Versioning and MFA Delete can be used to protect against accidental deletion.

Cross-region replication can be used to automatically copy new objects from a source bucket in one region to a target bucket in another region.

Pre-signed URLs grant time-limited permission to download objects and can be used to protect media and other web content from unauthorized “web scraping.”

Multipart upload can be used to upload large objects, and Range GETs can be used to download portions of an Amazon S3 object or Amazon Glacier archive.

Server access logs can be enabled on a bucket to track requestor, object, action, and response.

Amazon S3 event notifications can be used to send an Amazon SQS or Amazon SNS message or to trigger an AWS Lambda function when an object is created or deleted.

Amazon Glacier can be used as a standalone service or as a storage class in Amazon S3.

Amazon Glacier stores data in archives, which are contained in vaults. You can have up to 1,000 vaults, and each vault can store an unlimited number of archives.

Amazon Glacier vaults can be locked for compliance purposes.

**Exam Essentials**

**Know what amazon s3 is and what it is commonly used for.** Amazon S3 is secure, durable, and highly scalable cloud storage that can be used to store an unlimited amount of data in almost any format using a simple web services interface. Common use cases include backup and archive, content storage and distribution, big data analytics, static website hosting, cloud-native application hosting, and disaster recovery.

**Understand how object storage differs from block and file storage.** Amazon S3 cloud object storage manages data at the application level as objects using a REST API built on HTTP. Block storage manages data at the operating system level as numbered addressable blocks using protocols such as SCSI or Fibre Channel. File storage manages data as shared files at the operating system level using a protocol such as CIFS or NFS.

**Understand the basics of Amazon S3.** Amazon S3 stores data in objects that contain data and metadata. Objects are identified by a user-defined key and are stored in a simple flat folder called a bucket. Interfaces include a native REST interface, SDKs for many languages, an AWS CLI, and the AWS Management Console.

Know how to create a bucket; how to upload, download, and delete objects; how to make objects public; and how to open an object URL.

**Understand the durability, availability, and data consistency model of Amazon S3.** Amazon S3 standard storage is designed for 11 nines durability and four nines availability of objects over a year. Other storage classes differ. Amazon S3 is eventually consistent, but offers read-after-write consistency for PUTs to new objects.

**Know how to enable static website hosting on Amazon S3.** To create a static website on Amazon S3, you must create a bucket with the website hostname, upload your static content and make it public, enable static website hosting on the bucket, and indicate the index and error page objects.

**Know how to protect your data on Amazon S3.** Encrypt data in flight using HTTPS and at rest using SSE or client-side encryption. Enable versioning to keep multiple versions of an object in a bucket. Enable MFA Delete to protect against accidental deletion. Use ACLs Amazon S3 bucket policies and AWS IAM policies for access control. Use pre-signed URLs for time-limited download access. Use cross-region replication to automatically replicate data to another region.

**Know the use case for each of the Amazon S3 storage classes.** Standard is for general purpose data that needs high durability, high performance, and low latency access. Standard IA is for data that is less frequently accessed, but that needs the same performance and availability when accessed. RRS offers lower durability at lower cost for easily replicated data. Amazon Glacier is for storing rarely accessed archival data at lowest cost, when three- to five hour retrieval time is acceptable.

**Know how to use lifecycle configuration rules.** Lifecycle rules can be configured in the AWS Management Console or the APIs. Lifecycle configuration rules define actions to transition objects from one storage class to another based on time.

**Know how to use Amazon S3 event notifications.** Event notifications are set at the