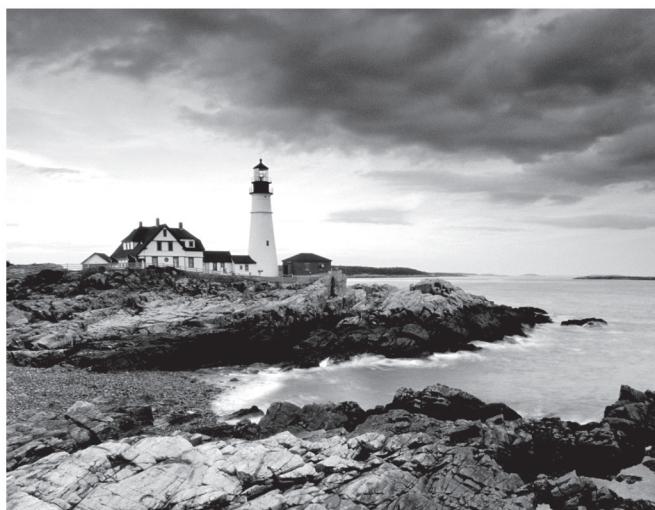


Official Google Cloud Certified

Associate Cloud Engineer

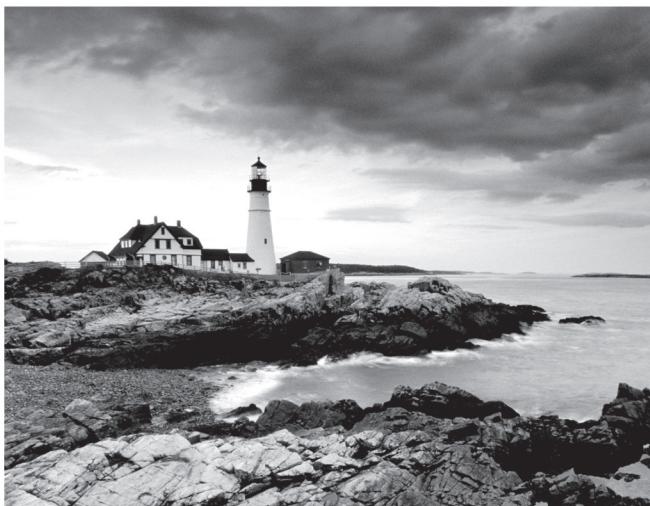
Study Guide



Official Google Cloud Certified

Associate Cloud Engineer

Study Guide



Dan Sullivan

Development Editor: Stephanie Barton
Technical Editors: Stacy Véronneau and Manjeet Dadyala
Google Technical Reviewers: Jake Bednard, Brian Rice, Teresa Hardy, Grace Mollison, Tanay Buddhdev, Richard Rose, Jasen Baker, Jim Rambo, Varsha Datta, Mylene Biddle, Evan Jones, Samar Bhat, Josh Koh, Jeff Sherman, Kuntal Mitra, Michael Arciola and Lisa Guinn
Senior Production Editor: Christine O'Connor
Copy Editor: Kim Wimpsett
Content Enablement and Operations Manager: Pete Gaughan
Production Manager: Kathleen Wisor
Associate Publisher: Jim Minatel
Book Designers: Judy Fung and Bill Gibson
Proofreader: Louise Watson, Word One New York
Indexer: Johnna VanHoose Dinse
Project Coordinator, Cover: Brent Savage
Cover Designer: Wiley
Cover Image: Getty Images Inc. / Jeremy Woodhouse

Copyright © 2019 by John Wiley & Sons, Inc., Indianapolis, Indiana

Published simultaneously in Canada

ISBN: 978-1-119-56441-6

ISBN: 978-1-119-56418-8 (ebk.)

ISBN: 978-1-119-56439-3 (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: 2019936130

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. Google Cloud and the Google Cloud logo are trademarks of Google LLC and/or its affiliates, in the United States and other countries, and may not be used without written permission. 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.

to Katherine

Acknowledgments

A book is a team effort—more so for this book than any I've written before.

I am fortunate to have had the opportunity to work with Jim Minatel, associate publisher at John Wiley & Sons, and Carole Jelen, VP of Waterside Productions. Carole and Jim shared their vision for this book and invited me into their endeavor. They have been through the entire, sometimes time-challenged, writing of this book. Their knowledge and experience led to an improved book over the one you might otherwise be reading.

I am especially grateful for Stephanie Barton's help shaping the manuscript into its finished form. Stephanie edited the text, deciphered awkward grammatical contortions, and helped me think through the pedagogy of question design.

Thank you to Christine O'Connor for shepherding this book through the production process, which had more moving parts than I could track. Thanks to Kim Wimpsett for getting the text into final shape and suitable for the public.

I appreciate the careful attention and close reading by the technical editors, whose efforts made the text more precise and accurate.

I am fortunate to work in a stimulating professional environment where the culture supports who we are as individuals and enables a level of collaboration I've never experienced before joining New Relic. Thank you to my director, Cathy Rotering, who has a talent for seeing what makes people tick and helping them get where they want to go.

Like any accomplishment that might look like my doing, this book is possible because of those closest to me: Meg; all five of my children, particularly James and Nicholas, who were my first readers; and most importantly my wife, Katherine. For the 35 years we've been together, Katherine has engaged life with joy. Her own demanding work in literary publishing and poetry is changing lives, but Katherine is always present for those she loves, especially me.

About the Author

Dan Sullivan is a principal engineer and software architect at New Relic. He specializes in streaming analytics, machine learning, and cloud computing. Dan is the author of *NoSQL for Mere Mortals* and several LinkedIn Learning courses on databases, data science, and machine learning. Dan has certifications from Google and AWS along with a Ph.D. in genetics and computational biology.

Contents at a Glance

<i>Introduction</i>	<i>xxi</i>	
<i>Assessment Test</i>	<i>xxxii</i>	
Chapter 1	Overview of Google Cloud Platform	1
Chapter 2	Google Cloud Computing Services	15
Chapter 3	Projects, Service Accounts, and Billing	39
Chapter 4	Introduction to Computing in Google Cloud	67
Chapter 5	Computing with Compute Engine Virtual Machines	91
Chapter 6	Managing Virtual Machines	117
Chapter 7	Computing with Kubernetes	145
Chapter 8	Managing Kubernetes Clusters	175
Chapter 9	Computing with App Engine	209
Chapter 10	Computing with Cloud Functions	225
Chapter 11	Planning Storage in the Cloud	241
Chapter 12	Deploying Storage in Google Cloud Platform	275
Chapter 13	Loading Data into Storage	309
Chapter 14	Networking in the Cloud: Virtual Private Clouds and Virtual Private Networks	337
Chapter 15	Networking in the Cloud: DNS, Load Balancing, and IP Addressing	361
Chapter 16	Deploying Applications with Cloud Launcher and Deployment Manager	383
Chapter 17	Configuring Access and Security	405
Chapter 18	Monitoring, Logging, and Cost Estimating	425
Appendix	Answers to Review Questions	463
<i>Index</i>	<i>503</i>	

Contents

<i>Introduction</i>	<i>xxi</i>
<i>Assessment Test</i>	<i>xxxii</i>
Chapter 1 Overview of Google Cloud Platform	1
Types of Cloud Services	2
Compute Resources	3
Storage	4
Networking	7
Specialized Services	8
Cloud Computing vs. Data Center Computing	8
Rent Instead of Own Resources	8
Pay-as-You-Go-for-What-You-Use Model	9
Elastic Resource Allocation	9
Specialized Services	10
Exam Essentials	10
Review Questions	11
Chapter 2 Google Cloud Computing Services	15
Computing Components of Google Cloud Platform	16
Computing Resources	17
Storage Components of Google Cloud Platform	21
Storage Resources	21
Databases	23
Networking Components of Google Cloud Platform	26
Networking Services	26
Identity Management	28
Development Tools	28
Additional Components of Google Cloud Platform	29
Management Tools	29
Specialized Services	30
Exam Essentials	31
Review Questions	34
Chapter 3 Projects, Service Accounts, and Billing	39
How GCP Organizes Projects and Accounts	40
GCP Resource Hierarchy	40
Organization Policies	43
Managing Projects	45

	Roles and Identities	47
	Roles in GCP	47
	Granting Roles to Identities	49
	Service Accounts	50
	Billing	51
	Billing Accounts	51
	Billing Budgets and Alerts	53
	Exporting Billing Data	54
	Enabling APIs	56
	Provisioning Stackdriver Workspaces	58
	Exam Essentials	60
	Review Questions	61
Chapter 4	Introduction to Computing in Google Cloud	67
	Compute Engine	68
	Virtual Machine Images	68
	Virtual Machines Are Contained in Projects	71
	Virtual Machines Run in a Zone and Region	72
	Users Need Privileges to Create Virtual Machines	73
	Preemptible Virtual Machines	74
	Custom Machine Types	76
	Use Cases for Compute Engine Virtual Machines	77
	App Engine	77
	Structure of an App Engine Application	77
	App Engine Standard and Flexible Environments	78
	Use Cases for App Engine	80
	Kubernetes Engine	81
	Kubernetes Functionality	82
	Kubernetes Cluster Architecture	82
	Kubernetes High Availability	83
	Kubernetes Engine Use Cases	84
	Cloud Functions	84
	Cloud Functions Execution Environment	84
	Cloud Functions Use Cases	85
	Summary	85
	Exam Essentials	86
	Review Questions	87
Chapter 5	Computing with Compute Engine Virtual Machines	91
	Creating and Configuring Virtual Machines with the Console	92
	Main Virtual Machine Configuration Details	95
	Additional Configuration Details	97

	Creating and Configuring Virtual Machines with Cloud SDK	103
	Installing Cloud SDK	103
	Cloud SDK on Mac OS	104
	Example Installation on Ubuntu Linux	104
	Creating a Virtual Machine with Cloud SDK	105
	Creating a Virtual Machine with Cloud Shell	106
	Basic Virtual Machine Management	107
	Starting and Stopping Instances	107
	Network Access to Virtual Machines	108
	Monitoring a Virtual Machine	109
	Cost of Virtual Machines	111
	Guidelines for Planning, Deploying, and Managing Virtual Machines	111
	Summary	112
	Exam Essentials	112
	Review Questions	113
Chapter 6	Managing Virtual Machines	117
	Managing Single Virtual Machine Instances	118
	Managing Single Virtual Machine Instances in the Console	118
	Managing a Single Virtual Machine Instance with Cloud Shell and the Command Line	130
	Introduction to Instance Groups	134
	Creating and Removing Instance Groups and Templates	134
	Instance Groups Load Balancing and Autoscaling	137
	Guidelines for Managing Virtual Machines	137
	Summary	138
	Exam Essentials	138
	Review Questions	140
Chapter 7	Computing with Kubernetes	145
	Introduction to Kubernetes Engine	146
	Kubernetes Cluster Architecture	146
	Kubernetes Objects	147
	Deploying Kubernetes Clusters	149
	Deploying Kubernetes Clusters Using Cloud Console	149
	Deploying Kubernetes Clusters Using Cloud Shell and Cloud SDK	153
	Deploying Application Pods	154
	Monitoring Kubernetes	157
	Summary	168
	Exam Essentials	168
	Review Questions	170

Chapter 8	Managing Kubernetes Clusters	175
	Viewing the Status of a Kubernetes Cluster	176
	Viewing the Status of Kubernetes Clusters Using Cloud Console	176
	Viewing the Status of Kubernetes Clusters Using Cloud SDK and Cloud Shell	184
	Adding, Modifying, and Removing Nodes	190
	Adding, Modifying, and Removing Nodes with Cloud Console	190
	Adding, Modifying, and Removing with Cloud SDK and Cloud Shell	191
	Adding, Modifying, and Removing Pods	192
	Adding, Modifying, and Removing Pods with Cloud Console	192
	Adding, Modifying, and Removing Pods with Cloud SDK and Cloud Shell	195
	Adding, Modifying, and Removing Services	196
	Adding, Modifying, and Removing Services with Cloud Console	196
	Adding, Modifying, and Removing Services with Cloud SDK and Cloud Shell	198
	Viewing the Image Repository and Image Details	199
	Viewing the Image Repository and Image Details with Cloud Console	199
	Viewing the Image Repository and Image Details with Cloud SDK and Cloud Shell	202
	Summary	203
	Exam Essentials	203
	Review Questions	204
Chapter 9	Computing with App Engine	209
	App Engine Components	210
	Deploying an App Engine Application	211
	Deploying an App Using Cloud Shell and SDK	211
	Scaling App Engine Applications	215
	Splitting Traffic between App Engine Versions	217
	Summary	218
	Exam Essentials	218
	Review Questions	220
Chapter 10	Computing with Cloud Functions	225
	Introduction to Cloud Functions	226
	Events, Triggers, and Functions	227
	Runtime Environments	227
	Cloud Functions Receiving Events from Cloud Storage	229

Deploying a Cloud Function for Cloud Storage Events Using Cloud Console	229
Deploying a Cloud Function for Cloud Storage Events Using gcloud Commands	231
Cloud Functions Receiving Events from Pub/Sub	233
Deploying a Cloud Function for Cloud Pub/Sub Events Using Cloud Console	233
Deploying a Cloud Function for Cloud Pub/Sub Events Using gcloud Commands	234
Summary	234
Exam Essentials	235
Review Questions	237
Chapter 11 Planning Storage in the Cloud	241
Types of Storage Systems	242
Cache	243
Persistent Storage	245
Object Storage	247
Storage Types When Planning a Storage Solution	253
Storage Data Models	254
Object: Cloud Storage	254
Relational: Cloud SQL, Cloud Spanner, and BigQuery	254
NoSQL: Datastore, Cloud Firestore, and Bigtable	261
Choosing a Storage Solution: Guidelines to Consider	268
Summary	269
Exam Essentials	270
Review Questions	271
Chapter 12 Deploying Storage in Google Cloud Platform	275
Deploying and Managing Cloud SQL	276
Creating and Connecting to a MySQL Instance	276
Creating a Database, Loading Data, and Querying Data	278
Backing Up MySQL in Cloud SQL	279
Deploying and Managing Datastore	283
Adding Data to a Datastore Database	283
Backing Up Datastore	284
Deploying and Managing BigQuery	285
Estimating the Cost of Queries in BigQuery	285
Viewing Jobs in BigQuery	286
Deploying and Managing Cloud Spanner	288
Deploying and Managing Cloud Pub/Sub	292
Deploying and Managing Cloud Bigtable	295
Deploying and Managing Cloud Dataproc	298

	Managing Cloud Storage	302
	Summary	303
	Exam Essentials	304
	Review Questions	305
Chapter 13	Loading Data into Storage	309
	Loading and Moving Data to Cloud Storage	310
	Loading and Moving Data to Cloud Storage Using the Console	310
	Loading and Moving Data to Cloud Storage Using the Command Line	314
	Importing and Exporting Data	315
	Importing and Exporting Data: Cloud SQL	315
	Importing and Exporting Data: Cloud Datastore	319
	Importing and Exporting Data: BigQuery	320
	Importing and Exporting Data: Cloud Spanner	325
	Importing and Exporting Data: Cloud Bigtable	327
	Importing and Exporting Data: Cloud Dataproc	329
	Streaming Data to Cloud Pub/Sub	330
	Summary	331
	Exam Essentials	332
	Review Questions	333
Chapter 14	Networking in the Cloud: Virtual Private Clouds and Virtual Private Networks	337
	Creating a Virtual Private Cloud with Subnets	338
	Creating a Virtual Private Cloud with Cloud Console	338
	Creating a Virtual Private Cloud with gcloud	342
	Creating a Shared Virtual Private Cloud Using gcloud	343
	Deploying Compute Engine with a Custom Network	344
	Creating Firewall Rules for a Virtual Private Cloud	347
	Structure of Firewall Rules	347
	Creating Firewall Rules Using Cloud Console	348
	Creating Firewall Rules Using gcloud	350
	Creating a Virtual Private Network	351
	Creating a Virtual Private Network Using Cloud Console	351
	Creating a Virtual Private Network Using gcloud	354
	Summary	355
	Exam Essentials	355
	Review Questions	357

Chapter 15	Networking in the Cloud: DNS, Load Balancing, and IP Addressing	361
Configuring Cloud DNS	362	
Creating DNS Managed Zones Using Cloud Console	362	
Creating a DNS Managed Zones Using gcloud	366	
Configuring Load Balancers	367	
Types of Load Balancers	367	
Configuring Load Balancers using Cloud Console	368	
Configuring Load Balancers using gcloud	374	
Managing IP Addresses	375	
Expanding CIDR Blocks	375	
Reserving IP Addresses	375	
Summary	377	
Exam Essentials	378	
Review Questions	379	
Chapter 16	Deploying Applications with Cloud Launcher and Deployment Manager	383
Deploying a Solution Using Cloud Launcher	384	
Browsing Cloud Launcher and Viewing Solutions	384	
Deploying Cloud Launcher Solutions	390	
Deploying an Application Using Deployment Manager	394	
Deployment Manager Configuration Files	394	
Deployment Manager Template Files	397	
Launching a Deployment Manager Template	398	
Summary	398	
Exam Essentials	399	
Review Questions	400	
Chapter 17	Configuring Access and Security	405
Managing Identity and Access Management	406	
Viewing Account Identity and Access Management	406	
Assignments	406	
Assigning Identity and Access Management Roles to Accounts and Groups	408	
Defining Custom Identity and Access Management Roles	411	
Managing Service Accounts	414	
Managing Service Accounts with Scopes	414	
Assigning a Service Account to a Virtual Machine Instance	416	
Viewing Audit Logs	417	
Summary	418	
Exam Essentials	418	
Review Questions	420	

Chapter 18	Monitoring, Logging, and Cost Estimating	425
Monitoring with Stackdriver	426	
Creating Alerts Based on Resource Metrics	427	
Creating Custom Metrics	437	
Logging with Stackdriver	438	
Configuring Log Sinks	438	
Viewing and Filtering Logs	441	
Viewing Message Details	444	
Using Cloud Diagnostics	446	
Overview of Cloud Trace	446	
Overview of Cloud Debug	448	
Viewing Google Cloud Platform Status	451	
Using the Pricing Calculator	453	
Summary	457	
Exam Essentials	458	
Review Questions	459	
Appendix	Answers to Review Questions	463
Chapter 1: Overview of Google Cloud Platform	464	
Chapter 2: Google Cloud Computing Services	466	
Chapter 3: Projects, Service Accounts, and Billing	468	
Chapter 4: Introduction to Computing in Google Cloud	470	
Chapter 5: Computing with Compute Engine Virtual Machines	472	
Chapter 6: Managing Virtual Machines	475	
Chapter 7: Computing with Kubernetes	477	
Chapter 8: Managing Kubernetes Clusters	479	
Chapter 9: Computing with App Engine	481	
Chapter 10: Computing with Cloud Functions	483	
Chapter 11: Planning Storage in the Cloud	485	
Chapter 12: Deploying Storage in Google Cloud Platform	487	
Chapter 13: Loading Data into Storage	490	
Chapter 14: Networking in the Cloud Virtual Private Clouds and Virtual Private Networks	492	
Chapter 15: Networking in the Cloud: DNS, Load Balancing, and IP Addressing	494	
Chapter 16: Deploying Applications with Cloud Launcher and Deployment Manager	496	
Chapter 17: Configuring Access and Security	498	
Chapter 18: Monitoring, Logging, and Cost Estimating	500	
<i>Index</i>		503

Introduction

Google Cloud Platform (GCP) is a leading public cloud that provides its users with some of the same software, hardware, and networking infrastructure used to power Google services. Businesses, organizations, and individuals can launch servers in minutes, store petabytes of data, and implement global virtual clouds with GCP. It includes an easy-to-use console interface, command-line tools, and application programming interfaces (APIs) for managing resources in the cloud. Users can work with general resources, such as virtual machines (VMs) and persistent disks, or opt for highly focused services for Internet of Things (IoT), machine learning, media, and other specialized domains.

Deploying and managing applications and services in GCP requires a clear understanding of the way Google structures user accounts and manages identities and access controls; you also need to understand the advantages and disadvantages of using various services. Certified Associate Cloud Engineers have demonstrated the knowledge and skills needed to deploy and operate infrastructure, services, and networks in Google Cloud.

This study guide is designed to help you understand GCP in depth so that you can meet the needs of those operating resources in Google Cloud. Yes, this book will, of course, help you pass the Associate Cloud Engineer certification exam, but this is not an exam cram guide. You will learn more than is required to pass the exam; you will understand how to meet the day-to-day challenges faced by cloud engineers, including choosing services, managing users, deploying and monitoring infrastructure, and helping map business requirements into cloud-based solutions.

Each chapter in this book covers a single topic and includes an “Exam Essentials” section that outlines key information you should know to pass the certification exam. There are also exercises to help you review and reinforce your understanding of the chapter’s topic. Sample questions are included at the end of each chapter so you can get a sense of the types of questions you will see on the exam. The book also includes flashcards and practice exams that cover all topics you’ll learn about with this guide.

What Does This Book Cover?

This book describes products and services in GCP. It does not include G Suite administration topics.

Chapter 1: Overview of Google Cloud Platform In the opening chapter, we look into the types of services provided by GCP, which include compute, storage, and networking services as well as specialized services, such as machine learning products. This chapter

also describes some of the key differences between cloud computing and data center or on-premise computing.

Chapter 2: Google Cloud Computing Services This chapter provides an overview of infrastructure services such as computing, storage, and networking. It introduces the concept of identity management and related services. It also introduces DevOps topics and tools for deploying and monitoring applications and resources. GCP includes a growing list of specialized services, such as machine learning and natural language processing services. Those are briefly discussed in this chapter. The chapter introduces Google Cloud's organizational structure with a look at regions and zones. The chapter concludes with a discussion of Cloud Launcher for deploying packaged applications.

Chapter 3: Projects, Service Accounts, and Billing One of the first things you will do when starting to work with GCP is to set up your accounts. In this chapter, you will learn how resources in accounts are organized into organizations, folders, and projects. You will learn how to create and edit these structures. You will also see how to enable APIs for particular projects as well as manage user identities and their access controls. This chapter describes how to create billing accounts and link them to projects. You will also learn how to create budgets and define billing alerts to help you manage costs. Finally, the chapter describes how to create Stackdriver accounts, which are used as part of the monitoring system in GCP.

Chapter 4: Introduction to Computing in Google Cloud In this chapter, you will see the variety of options available for running applications and services in GCP. Options include Compute Engine, which provides VMs running Linux or Windows operating systems. App Engine is a platform as a service (PaaS) option that allows developers to run their applications without having to concern themselves with managing VMs. If you will be running multiple applications and services, you may want to take advantage of containers, which are a lightweight alternative to VMs. You will learn about containers and how to manage them with Kubernetes Engine. This chapter also introduces Cloud Functions, which is for event-driven, short-running tasks such as triggering the processing of an image loaded into Cloud Storage. You will also learn about Firebase, a set of services well suited for providing backend infrastructure to mobile applications.

Chapter 5: Computing with Compute Engine Virtual Machines In this chapter, you will learn how to configure VMs, including selecting CPU, memory, storage options, and operating system images. You will learn how to use GCP Console and Cloud Shell to work with VMs. In addition, you will see how to install the command-line interface and SDK, which you will use to start and stop VMs. The chapter will also describe how to enable network access to VMs.

Chapter 6: Managing Virtual Machines In the previous chapter, you learned how to create VMs, and in this chapter you will learn how to manage individual and groups of VMs. You will start by managing a single instance of a VM using the GCP console and then perform the same operations using Cloud Shell and the command line. You

will also learn how to view currently running VMs. Next, you'll learn about instance groups, which allow you to create sets of VMs that you can manage as a single unit. In the section on instance groups, you will learn the difference between managed and unmanaged instance groups. You will also learn about preemptible instances, which are low-cost VMs that may be shut down by Google. You will learn about the cost-benefit trade-offs of preemptible instances. Finally, the chapter closes with guidelines for managing VMs.

Chapter 7: Computing with Kubernetes This chapter introduces Kubernetes Engine, Google's managed Kubernetes service. Kubernetes is a container orchestration platform created and released as open source by Google. In this chapter, you will learn the basics of containers, container orchestration, and the Kubernetes architecture. The discussion will include an overview of Kubernetes objects such as pods, services, volumes, and namespaces, as well as Kubernetes controllers such as ReplicaSets, deployments, and jobs.

Next, the chapter turns to deploying a Kubernetes cluster using GCP console, Cloud Shell, and SDK. You will also see how to deploy pods, which includes downloading an existing Docker image, building a Docker image, creating a pod, and then deploying an application to the Kubernetes cluster. Of course, you will need to know how to monitor a cluster of servers. This chapter provides a description of how to set up monitoring and logging with Stackdriver, which is Google's application, service, container, and infrastructure monitoring service.

Chapter 8: Managing Kubernetes Clusters In this chapter you will learn the basics of managing a Kubernetes cluster, including viewing the status of the cluster, viewing the contents of the image repository, viewing details about images in the repository, and adding, modifying, and removing nodes, pods, and services. As in the chapter on managing VMs, in this chapter you will learn how to perform management operations with the three management tools: GCP console, Cloud Shell, and SDK. The chapter concludes with a discussion of guidelines and good practices for managing a Kubernetes cluster.

Chapter 9: Computing with App Engine Google App Engine is Google's PaaS offering. You will learn about App Engine components such as applications, services, versions, and instances. The chapter also covers how to define configuration files and specify dependencies of an application. In this chapter, you will learn how to view App Engine resources using GCP console, Cloud Shell, and SDK. The chapter also describes how to distribute workload by adjusting traffic with splitting parameters. You will also learn about autoscaling in App Engine.

Chapter 10: Computing with Cloud Functions Cloud Functions is for event-driven, serverless computations. This chapter introduces Cloud Functions, including using it to receive events, evoke services, and return results. Next, you'll see use cases for Cloud Functions, such as integrating with third-party APIs and event-driven processing. You will learn about Google's Pub/Sub service for publication- and subscription-based processing and how to use Cloud Functions with Pub/Sub. Cloud Functions are well suited to respond to events in Cloud Storage. The chapter describes Cloud Storage events and how to use Cloud

Functions to receive and respond to those events. You will learn how to use Stackdriver to monitor and log details of Cloud Function executions. Finally, the chapter concludes with a discussion of guidelines for using and managing Cloud Functions.

Chapter 11: Planning Storage in the Cloud Having described various compute options in GCP, it is time to turn your attention to storage. This chapter describes characteristics of storage systems, such as their time to access, persistence, and data model. In this chapter, you will learn about differences between caches, persistent storage, and archival storage. You will learn about the cost-benefit trade-offs of using regional and multiregional persistent storage and using nearline versus coldline archival storage. The chapter includes details on the various GCP storage options, including Cloud Storage for blob storage; Cloud SQL and Spanner for relational data; Datastore, Bigtable, and BigQuery for NoSQL storage; and Cloud Firebase for mobile application data. The chapter includes detailed guidance on choosing a data store based on requirements for consistency, availability, transaction support, cost, latency, and support for different read/write patterns.

Chapter 12: Deploying Storage in Google Cloud Platform In this chapter, you will learn how to create databases, add data, list records, and delete data from each of GCP's storage systems. The chapter starts by introducing Cloud SQL, a managed database service that offers MySQL and PostgreSQL managed instances. You will also learn how to create databases in Cloud Datastore, BigQuery, Bigtable, and Spanner. Next, you will turn your attention to Cloud Pub/Sub for storing data in message queues, followed by a discussion of Cloud Dataproc, a managed Hadoop and Spark cluster service, for processing big data sets. In the next section, you will learn about Cloud Storage for objects. The chapter concludes with guidance on how to choose a data store for a particular set of requirements.

Chapter 13: Loading Data into Storage There are a variety of ways of getting data into GCP. This chapter describes how to use the command-line SDK to load data into Cloud SQL, Cloud Storage, Datastore, BigQuery, BigTable, and Dataproc. It will also describe bulk importing and exporting from those same services. Next, you will learn about two common data loading patterns: moving data from Cloud Storage and streaming data to Cloud Pub/Sub.

Chapter 14: Networking in the Cloud: Virtual Private Clouds and Virtual Private Networks In this chapter, you'll turn your attention to networking with an introduction to basic networking concepts, including the following:

- IP addresses
- CIDR blocks
- Networks and subnetworks
- Virtual private clouds (VPCs)
- Routing and rules
- Virtual private networks (VPNs)
- Cloud DNS

- Cloud routers
- Cloud interconnect
- External peering

After being introduced to key networking concepts, you will learn how to create a VPC. Specifically, this will include defining a VPC, specifying firewall rules, creating a VPN, and working with load balancers. You will learn about different types of load balancers and when to use them.

Chapter 15: Networking in the Cloud: DNS, Load Balancing, and IP Addressing In this chapter, you will learn about common network management tasks such as defining subnetworks, adding subnets to a VPC, managing CIDR blocks, and reserving IP addresses. You will learn how to perform each of these tasks using Cloud Console, Cloud Shell, and Cloud SDK.

Chapter 16: Deploying Applications with Cloud Launcher and Deployment Manager Google Cloud Launcher is GCP’s marketplace of preconfigured stacks and services. This chapter introduces Cloud Launcher and describes some applications and services currently available. You will learn how to browse Cloud Launcher, deploy applications from Cloud Launcher, and shut down Cloud Launcher applications. The chapter will also discuss Deployment Manager templates to automate the deployment of an application and launch a Deployment Manager template to provision GCP resources and configure an application automatically.

Chapter 17: Configuring Access and Security This chapter introduces identity management. In particular, you will learn about identities, roles, and assigning and removing identity roles. This chapter also introduces service accounts and how to create them, assign them to VMs, and work with them across projects. You will also learn how to view audit logs for projects and services. The chapter concludes with guidelines for configuring access control security.

Chapter 18: Monitoring, Logging, and Cost Estimating In the final chapter, we will discuss Stackdriver alerts, logging, distributed tracing, and application debugging. Each of the corresponding GCP services is designed to enable more efficient, functional, and reliable services. The chapter concludes with a review of the Pricing Calculator, which is helpful for estimating the cost of resources in GCP.

Interactive Online Learning Environment and TestBank

Studying the material in the *Official Google Certified Associate Cloud Engineer Study Guide* is an important part of preparing for the Associate Cloud Engineer certification exam, but we provide additional tools to help you prepare. The online TestBank will help you understand the types of questions that will appear on the certification exam.

The sample tests in the TestBank include all the questions in each chapter as well as the questions from the assessment test. In addition, there are two practice exams with 50 questions each. You can use these tests to evaluate your understanding and identify areas that may require additional study.

The flashcards in the TestBank will push the limits of what you should know for the certification exam. There are 100 questions provided in digital format. Each flashcard has one question and one correct answer.

The online glossary is a searchable list of key terms introduced in this exam guide that you should know for the Associate Cloud Engineer certification exam.

To start using these to study for the Google Certified Associate Cloud Engineer exam, go to www.wiley.com/go/sybextestprep and register your book to receive your unique PIN. Once you have the PIN, return to www.wiley.com/go/sybextestprep, find your book and click Register or Login, and follow the link to register a new account or add this book to an existing account.

Exam Objectives

The Associate Cloud Engineer certification is designed for people who create, deploy, and manage enterprise applications and infrastructure in GCP. An Associate Cloud Engineer is comfortable working with Cloud Console, Cloud Shell, and Cloud SDK. Such individuals also understand products offered as part of GCP and their appropriate use cases.

The exam will test your knowledge of the following:

- Planning a cloud solution using one or more GCP services
- Creating a cloud environment for an organization
- Deploying applications and infrastructure
- Using monitoring and logging to ensure availability of cloud solutions
- Setting up identity management, access controls, and other security measures

Objective Map

The following are specific objectives defined by Google at <https://cloud.google.com/certification/guides/cloud-engineer/>.

Section 1: Setting up a cloud solution environment

1.1 Setting up cloud projects and accounts. Activities include:

- Creating projects
- Assigning users to predefined IAM (Identity and Access Management) roles within a project
- Linking users to G Suite identities

- Enabling APIs within projects
- Provisioning one or more Stackdriver accounts

1.2 Managing billing configuration. Activities include:

- Creating one or more billing accounts
- Linking projects to a billing account
- Establishing billing budgets and alerts
- Setting up billing exports to estimate daily/monthly charges

1.3 Installing and configuring the command-line interface (CLI), specifically Cloud SDK (e.g., setting the default project)

Section 2: Planning and configuring a cloud solution

2.1 Planning and estimating GCP product use using the Pricing Calculator

2.2 Planning and configuring compute resources. Considerations include:

- Selecting appropriate compute choices for a given workload (e.g., Compute Engine, Kubernetes Engine, App Engine)
- Using preemptible VMs and custom machine types as appropriate

2.3 Planning and configuring data storage options. Considerations include:

- Product choice (e.g., Cloud SQL, BigQuery, Cloud Spanner, Cloud Bigtable)
- Choosing storage options (e.g., Regional, Multiregional, Nearline, Coldline)

2.4 Planning and configuring network resources. Tasks include:

- Differentiating load balancing options
- Identifying resource locations in a network for availability
- Configuring Cloud DNS

Section 3: Deploying and implementing a cloud solution

3.1 Deploying and implementing Compute Engine resources. Tasks include:

- Launching a compute instance using Cloud Console and Cloud SDK (gcloud) (e.g., assign disks, availability policy, SSH keys)
- Creating an autoscaled managed instance group using an instance template
- Generating/uploading a custom SSH key for instances
- Configuring a VM for Stackdriver monitoring and logging
- Assessing compute quotas and requesting increases
- Installing the Stackdriver Agent for monitoring and logging

3.2 Deploying and implementing Kubernetes Engine resources. Tasks include:

- Deploying a Kubernetes Engine cluster
- Deploying a container application to Kubernetes Engine using pods
- Configuring Kubernetes Engine application monitoring and logging

3.3 Deploying and implementing App Engine and Cloud Functions resources. Tasks include:

- Deploying an application to App Engine (e.g., scaling configuration, versions, and traffic splitting)
- Deploying a Cloud Function that receives Google Cloud events (e.g., Cloud Pub/Sub events, Cloud Storage object change notification events)

3.4 Deploying and implementing data solutions. Tasks include:

- Initializing data systems with products (e.g., Cloud SQL, Cloud Datastore, BigQuery, Cloud Spanner, Cloud Pub/Sub, Cloud Bigtable, Cloud Dataproc, Cloud Storage)
- Loading data (e.g., command-line upload, API transfer, import/export, load data from Cloud Storage, streaming data to Cloud Pub/Sub)

3.5 Deploying and implementing networking resources. Tasks include:

- Creating a VPC with subnets (e.g., custom-mode VPC, shared VPC)
- Launching a Compute Engine instance with custom network configuration (e.g., internal-only IP address, Google private access, static external and private IP address, network tags)
- Creating ingress and egress firewall rules for a VPC (e.g., IP subnets, tags, service accounts)
- Creating a VPN between a Google VPC and an external network using Cloud VPN
- Creating a load balancer to distribute application network traffic to an application (e.g., global HTTP(S) load balancer, global SSL proxy load balancer, global TCP proxy load balancer, regional network load balancer, regional internal load balancer)

3.6 Deploying a Solution using Cloud Launcher. Tasks include:

- Browsing the Cloud Launcher catalog and viewing solution details
- Deploying a Cloud Launcher marketplace solution

3.7 Deploying an Application using Deployment Manager. Tasks include:

- Developing Deployment Manager templates to automate deployment of an application
- Launching a Deployment Manager template to provision GCP resources and configure an application automatically

Section 4: Ensuring successful operation of a cloud solution

4.1 Managing Compute Engine resources. Tasks include:

- Managing a single VM instance (e.g., start, stop, edit configuration, or delete an instance)
- SSH/RDP to the instance
- Attaching a GPU to a new instance and installing CUDA libraries
- Viewing current running VM inventory (instance IDs, details)
- Working with snapshots (e.g., create a snapshot from a VM, view snapshots, delete a snapshot)
- Working with images (e.g., create an image from a VM or a snapshot, view images, delete an image)
- Working with instance groups (e.g., set autoscaling parameters, assign an instance template, create an instance template, remove an instance group)
- Working with management interfaces (e.g., Cloud Console, Cloud Shell, Cloud SDK)

4.2 Managing Kubernetes Engine resources. Tasks include:

- Viewing current running cluster inventory (nodes, pods, services)
- Browsing the container image repository and viewing container image details
- Working with nodes (e.g., add, edit, or remove a node)
- Working with pods (e.g., add, edit, or remove pods)
- Working with services (e.g., add, edit, or remove a service)
- Working with management interfaces (e.g., Cloud Console, Cloud Shell, Cloud SDK)

4.3 Managing App Engine resources. Tasks include:

- Adjusting application traffic splitting parameters
- Setting scaling parameters for autoscaling instances
- Working with management interfaces (e.g., Cloud Console, Cloud Shell, Cloud SDK)

4.4 Managing data solutions. Tasks include:

- Executing queries to retrieve data from data instances (e.g., Cloud SQL, BigQuery, Cloud Spanner, Cloud Datastore, Cloud Bigtable, Cloud Dataproc)
- Estimating costs of a BigQuery query
- Backing up and restoring data instances (e.g., Cloud SQL, Cloud Datastore, Cloud Dataproc)
- Reviewing job status in Cloud Dataproc or BigQuery
- Moving objects between Cloud Storage buckets

- Converting Cloud Storage buckets between storage classes
- Setting object lifecycle management policies for Cloud Storage buckets
- Working with management interfaces (e.g., Cloud Console, Cloud Shell, Cloud SDK)

4.5 Managing networking resources. Tasks include:

- Adding a subnet to an existing VPC
- Expanding a CIDR block subnet to have more IP addresses
- Reserving static external or internal IP addresses
- Working with management interfaces (e.g., Cloud Console, Cloud Shell, Cloud SDK)

4.6 Monitoring and logging. Tasks include:

- Creating Stackdriver alerts based on resource metrics
- Creating Stackdriver custom metrics
- Configuring log sinks to export logs to external systems (e.g., on premise or BigQuery)
- Viewing and filtering logs in Stackdriver
- Viewing specific log message details in Stackdriver
- Using cloud diagnostics to research an application issue (e.g., viewing Cloud Trace data, using Cloud Debug to view an application point in time)
- Viewing GCP status
- Working with management interfaces (e.g., Cloud Console, Cloud Shell, Cloud SDK)

Section 5: Configuring access and security

5.1 Managing Identity and Access Management. Tasks include:

- Viewing account IAM assignments
- Assigning IAM roles to accounts or Google Groups
- Defining custom IAM roles

5.2 Managing service accounts. Tasks include:

- Managing service accounts with limited scopes
- Assigning a service account to VM instances
- Granting access to a service account in another project

5.3 Viewing audit logs for project and managed services

Cloud Computing Components

- Compute resources
- Storage
- Networking
- Specialized services

Difference between Cloud Computing and Data Center Computing

- Rent instead of own resources
- Pay-as-you-go-for-what-you-use model
- Elastic resource allocation
- Specialized services

Assessment Test

1. Instance templates are used to create a group of identical VMs. The instance templates include:
 - A. Machine type, boot disk image or container image, zone, and labels
 - B. Cloud Storage bucket definitions
 - C. A load balancer description
 - D. App Engine configuration file
2. The command-line command to create a Cloud Storage bucket is:
 - A. gcloud mb
 - B. gsutil mb
 - C. gcloud mkbucket
 - D. gsutil mkbucket
3. Your company has an object management policy that requires that objects stored in Cloud Storage be migrated from regional storage to nearline storage 90 days after the object is created. The most efficient way to do this is to:
 - A. Create a cloud function to copy objects from regional storage to nearline storage.
 - B. Set the `MigrateObjectAfter` property on the stored object to 90 days.
 - C. Copy the object to persistent storage attached to a VM and then copy the object to a bucket created on nearline storage.
 - D. Create a lifecycle management configuration policy specifying an age of 90 days and `SetStorageClass` as nearline.
4. An education client maintains a site where users can upload videos, and your client needs to assure redundancy for the files; therefore, you have created two buckets for Cloud Storage. Which command do you use to synchronize the contents of the two buckets?
 - A. gsutil rsync
 - B. gcloud cp sync
 - C. gcloud rsync
 - D. gsutil cp sync
5. VPCs are _____ resources.
 - A. Regional
 - B. Zonal
 - C. Global
 - D. Subnet

- 6.** A remote component in your network has failed, which results in a transient network error. When you submit a `gsutil` command, it fails because of a transient error. By default, the command will:
- A.** Terminate and log a message to Stackdriver
 - B.** Retry using a truncated binary exponential back-off strategy
 - C.** Prompt the user to decide to retry or quit
 - D.** Terminate and log a message to Cloud Shell
- 7.** All of the following are components of firewall rules except which one?
- A.** Direction of traffic
 - B.** Action on match
 - C.** Time to live (TTL)
 - D.** Protocol
- 8.** Adding virtual machines to an instance group can be triggered in an autoscaling policy by all of the following, except which one?
- A.** CPU utilization
 - B.** Stackdriver metrics
 - C.** IAM policy violation
 - D.** Load balancing serving capacity
- 9.** Your company's finance department is developing a new account management application that requires transactions and the ability to perform relational database operations using fully compliant SQL. Data store options in GCP include:
- A.** Spanner and Cloud SQL
 - B.** Datastore and Bigtable
 - C.** Spanner and Cloud Storage
 - D.** Datastore and Cloud SQL
- 10.** The marketing department in your company wants to deploy a web application but does not want to have to manage servers or clusters. A good option for them is:
- A.** Compute Engine
 - B.** Kubernetes Engine
 - C.** App Engine
 - D.** Cloud Functions
- 11.** Your company is building an enterprise data warehouse and wants SQL query capabilities over petabytes of data, but does not want to manage servers or clusters. A good option for them is:
- A.** Cloud Storage
 - B.** BigQuery
 - C.** Bigtable
 - D.** Datastore

- 12.** You have been hired as a consultant to a startup in the Internet of Things (IoT) space. The startup will stream large volumes of data into GCP. The data needs to be filtered, transformed, and analyzed before being stored in GCP Datastore. A good option for the stream processing component is:
- A.** Dataproc
 - B.** Cloud Dataflow
 - C.** Cloud Endpoints
 - D.** Cloud Interconnect
- 13.** Preemptible virtual machines may be shut down at any time but will always be shut down after running:
- A.** 6 hours
 - B.** 12 hours
 - C.** 24 hours
 - D.** 48 hours
- 14.** You have been tasked with designing an organizational hierarchy for managing departments and their cloud resources. What organizing components are available in GCP?
- A.** Organization, folders, projects
 - B.** Buckets, directories, subdirectories
 - C.** Organizations, buckets, projects
 - D.** Folders, buckets, projects
- 15.** During an incident that has caused an application to fail, you suspect some resource may not have appropriate roles granted. The command to list roles granted to a resource is:
- A.** gutil iam list-grantable-roles
 - B.** gcloud iam list-grantable-roles
 - C.** gcloud list-grantable-roles
 - D.** gcloud resources grantable-roles
- 16.** The availability of CPU platforms can vary between zones. To get a list of all CPU types available in a particular zone, you should use:
- A.** gcloud compute zones describe
 - B.** gcloud iam zones describe
 - C.** gutil zones describe
 - D.** gcloud compute regions list
- 17.** To create a custom role, a user must possess which role?
- A.** iam.create
 - B.** compute.roles.create
 - C.** iam.roles.create
 - D.** Compute.roles.add

- 18.** You have been asked to create a network with 1,000 IP addresses. In the interest of minimizing unused IP addresses, which CIDR suffix would you use to create a network with at least 1,000 addresses but no more than necessary?
- A.** /20
 - B.** /22
 - C.** /28
 - D.** /32
- 19.** A team of data scientists have asked for your help setting up an Apache Spark cluster. You suggest they use a managed GCP service instead of managing a cluster themselves on Compute Engine. The service they would use is:
- A.** Cloud Dataproc
 - B.** Cloud Dataflow
 - C.** Cloud Hadoop
 - D.** BigQuery
- 20.** You have created a web application that allows users to upload files to Cloud Storage. When files are uploaded, you want to check the file size and update the user's total storage used in their account. A serverless option for performing this action on load is:
- A.** Cloud Dataflow
 - B.** Cloud Dataproc
 - C.** Cloud Storage
 - D.** Cloud Functions
- 21.** Your company has just started using GCP, and executives want to have a dedicated connection from your data center to the GCP to allow for large data transfers. Which networking service would you recommend?
- A.** Google Cloud Carrier Internet Peering
 - B.** Google Cloud Interconnect – Dedicated
 - C.** Google Cloud Internet Peering
 - D.** Google Cloud DNS
- 22.** You want to have GCP manage cryptographic keys, so you've decided to use Cloud Key Management Services. Before you can start creating cryptographic keys, you must:
- A.** Enable Google Cloud Key Management Service (KMS) API and set up billing
 - B.** Enable Google Cloud KMS API and create folders
 - C.** Create folders and set up billing
 - D.** Give all users grantable roles to create keys

- 23.** In Kubernetes Engine, a node pool is:
- A.** A subset of nodes across clusters
 - B.** A set of VMs managed outside of Kubernetes Engine
 - C.** A set of preemptible VMs
 - D.** A subset of node instances within a cluster that all have the same configuration
- 24.** The GCP service for storing and managing Docker containers is:
- A.** Cloud Source Repositories
 - B.** Cloud Build
 - C.** Container Registry
 - D.** Docker Repository
- 25.** Code for Cloud Functions can be written in:
- A.** Node.js and Python
 - B.** Node.js, Python, and Go
 - C.** Python and Go
 - D.** Python and C

Answers to Assessment Test

1. A. Machine type, boot disk image or container image, zone, and labels are all configuration parameters or attributes of a VM and therefore would be included in an instance group configuration that creates those VMs.
2. B. gsutil is the command line for accessing and manipulating Cloud Storage from the command line. mb is the specific command for creating, or making, a bucket.
3. D. The lifecycle configuration policy allows administrators to specify criteria for migrating data to other storage systems without having to concern themselves with running jobs to actually execute the necessary steps. The other options are inefficient or do not exist.
4. A. gsutil is the command-line tool for working with Cloud Storage. rsync is the specific command in gsutil for synchronizing buckets.
5. C. Google operates a global network, and VPCs are resources that can span that global network.
6. B. gcloud by default will retry a failed network operation and will wait a long time before each retry. The time to wait is calculated using a truncated binary exponential back-off strategy.
7. C. Firewall rules do not have TTL parameters. Direction of traffic, action on match, and protocol are all components of firewall rules.
8. C. IAM policy violations do not trigger changes in the size of clusters. All other options can be used to trigger a change in cluster size.
9. A. Only Spanner and Cloud SQL databases support transactions and have a SQL interface. Datastore has transactions but does not support fully compliant SQL; it has a SQL-like query language. Cloud Storage does not support transactions or SQL.
10. C. App Engine is a PaaS that allows developers to deploy full applications without having to manage servers or clusters. Compute Engine and Kubernetes Engine require management of servers. Cloud Functions is suitable for short-running Node.js or Python functions but not full applications.
11. B. BigQuery is designed for petabyte-scale analytics and provides a SQL interface.
12. B. Cloud Dataflow allows for stream and batch processing of data and is well suited for this kind of ETL work. Dataproc is a managed Hadoop and Spark service that is used for big data analytics. Cloud Endpoints is an API service, and Cloud Interconnect is a network service.
13. C. If a preemptible machine has not been shut down within 24 hours, Google will stop the instance.
14. A. Organizations, folders, and projects are the components used to manage an organizational hierarchy. Buckets, directories, and subdirectories are used to organize storage.

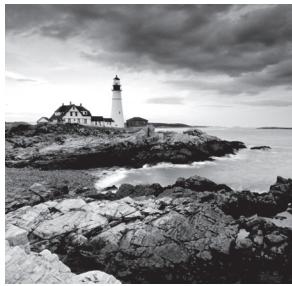
- 15.** B. `gcloud` is the command-line tool for working with IAM, and `list-grantable-roles` is the correct command.
- 16.** A. `gcloud` is the command-line tool for manipulating compute resources, and `zones describe` is the correct command.
- 17.** C. `iam.roles.create` is correct; the other roles do not exist.
- 18.** B. The `/22` suffix produces 1,022 usable IP addresses.
- 19.** A. Cloud Dataproc is the managed Spark service. Cloud Dataflow is for stream and batch processing of data, BigQuery is for analytics, and Cloud Hadoop is not a GCP service.
- 20.** D. Cloud Functions responds to events in Cloud Storage, making them a good choice for taking an action after a file is loaded.
- 21.** B. Google Cloud Interconnect – Dedicated is the only option for a dedicated connection between a customer’s data center and a Google data center.
- 22.** A. Enabling the Google Cloud KMS API and setting up billing are steps common to using GCP services.
- 23.** D. A node pool is a subset of node instances within a cluster that all have the same configuration.
- 24.** C. The GCP service for storing and managing Docker containers is Container Registry. Cloud Build is for creating images. The others are not GCP services.
- 25.** A. Node.js 6, Node.js 8, and Python are the languages supported by Cloud Functions.

Chapter 1

Overview of Google Cloud Platform

THIS CHAPTER COVERS THE FOLLOWING OBJECTIVE OF THE GOOGLE ASSOCIATE CLOUD ENGINEER CERTIFICATION EXAM:

- ✓ 1.0 Setting up a Cloud Solution Environment



Google Cloud Platform (GCP) is a public cloud service that offers some of the same technologies used by Google to deliver its own products. This chapter describes the most important components of GCP and discusses how it differs from on-premise data center-based computing.

Types of Cloud Services

Public cloud providers such as Google, Amazon, and Microsoft offer a range of services for deploying computing, storage, networking, and other infrastructures to run a wide array of business services and applications. Some cloud users are new companies that start in the cloud. They have never owned their own hardware and software. Other cloud customers are enterprises with multiple data centers that use public clouds to supplement their data centers. These different kinds of users have different requirements.

A company that starts on the cloud can choose services that best fit its application and architectural needs without having to consider existing infrastructure. For example, a startup could use GCP's Cloud Identity and Access Management services for all authentication and authorization needs. A company that has already invested in a Microsoft Active Directory solution for identity management may want to leverage that system instead of working solely with the cloud's identity management system. This can lead to additional work to integrate the two systems and keep them synchronized.

Another area of concern for enterprises with their own infrastructure is establishing and maintaining a secure network between their on-premise resources and their public cloud resources. If there will be high-volume network traffic between the on-premise systems and the public cloud, the enterprise may need to invest in dedicated networking between its data center and a facility of the public cloud provider. If the volume of traffic does not justify the cost of a dedicated connection between facilities, then the company may use a virtual private network that runs over the public Internet. This requires additional network design and management that a company that is solely in the cloud would not have to address.

Public cloud providers offer services that fall into four broad categories.

- Compute resources
- Storage
- Networking
- Specialized services such as Machine Learning Services

Cloud customers typically make use of services in more than one of these categories.

Compute Resources

Computing resources come in a variety of forms in public clouds.

Virtual Machines

Virtual Machines are a basic unit of computing resources and a good starting point for experimenting with the cloud. After you create an account with a cloud provider and provide billing information, you can use a portal or command-line tools to create VMs. Google Cloud Platform offers a variety of preconfigured VMs with varying numbers of vCPUs and amounts of memory. You can also create a custom configuration if the preconfigured offerings do not meet your needs.

Once you create a VM, you can log into it and administer it as you like. You have full access to the VM, so you can configure file systems, add persistent storage, patch the operating system, or install additional packages. You decide what to run on the VM, who else will have access to it, and when to shut down the VM. A VM that you manage is like having a server in your office that you have full administrator rights to.

You can, of course, create multiple VMs running different operating systems and applications. GCP also provides services, such as load balancers, that provide a single access point to a distributed back end. This is especially useful when you need to have high availability for your application. If one of the VMs in a cluster fails, the workload can be directed to the other VMs in the cluster. Autoscalers can add or remove VMs from the cluster based on the workload. This is called *autoscaling*. This helps both control cost by not running more VMs than needed and ensure that sufficient computing capacity is available when workloads increase.

Managed Kubernetes Clusters

Google Cloud Platform gives you all the tools you need to create and manage clusters of servers. Many cloud users would rather focus on their applications and not the tasks needed to keep a cluster of servers up and running. For those users, managed clusters are a good option.

Managed clusters make use of containers. A container is like a lightweight VM that isolates processes running in one container from processes running in another container on the same server. In a managed cluster, you can specify the number of servers you would like to run and the containers that should run on them. You can also specify autoscaling parameters to optimize the number of containers running.

In a managed cluster, the health of containers is monitored for you. If a container fails, the cluster management software will detect it and start another container.

Containers are good options when you need to run applications that depend on multiple micro services running in your environment. The services are deployed through containers, and the cluster management service takes care of monitoring, networking, and some security management tasks.

Serverless Computing

Both VMs and managed Kubernetes clusters require some level of effort to configure and administer computing resources. Serverless computing is an approach that allows developers and application administrators to run their code in a computing environment that does not require setting up VMs or Kubernetes clusters.

Google Cloud Platform has two serverless computing options: App Engine and Cloud Functions. App Engine is used for applications and containers that run for extended periods of time, such as a website backend, point-of-sale system, or custom business application. Cloud Functions is a platform for running code in response to an event, such as uploading a file or adding a message to a message queue. This serverless option works well when you need to respond to an event by running a short process coded in a function or by calling a longer-running application that might be running on a VM, managed cluster, or App Engine.

Storage

Public clouds offer a few types of storage services that are useful for a wide range of application requirements. These types include the following:

- Object storage
- File storage
- Block storage
- Caches

Enterprise users of cloud services will often use a combination of these services.

Object Storage

Object storage is a system that manages the use of storage in terms of objects or blobs. Usually these objects are files, but it is important to note that the files are not stored in a conventional file system. Objects are grouped into buckets. Each object is individually addressable, usually by a URL.

Object storage is not limited by the size of disks or solid-state drives (SSDs) attached to a server. Objects can be uploaded without concern for the amount of space available on a disk. Multiple copies of objects are stored to improve availability and durability. In some cases, copies of objects may be stored in different regions to ensure availability even if a region becomes inaccessible.

Another advantage of object storage is that it is serverless. There is no need to create VMs and attach storage to them. Google Cloud Platform's object storage, called Cloud Storage, is accessible from servers running in GCP as well as from other devices with Internet access.

Access controls can be applied at the object level. This allows users of cloud storage to control which users can access and update objects.

File Storage

File storage services provide a hierarchical storage system for files. File systems storage provides network shared file systems. Google Cloud Platform has a file storage service called Cloud Filestore, which is based on the Network File System (NFS) storage system.

File storage is suitable for applications that require operating system-like file access to files. The file storage system decouples the file system from specific VMs. The file system, its directories, and its files exist independent of VMs or applications that may access those files.

Block Storage

Block storage uses a fixed-size data structure called a *block* to organize data. Block storage is commonly used in ephemeral and persistent disks attached to VMs. With a block storage system, you can install file systems on top of the block storage, or you can run applications that access blocks directly. Some relational databases can be designed to access blocks directly rather working through file systems.

In Linux file systems, 4KB is a common block size. Relational databases often write directly to blocks, but they often use larger sizes, such as 8KB or more.

Block storage is available on disks that are attached to VMs in Google Cloud Platform. Block storage can be either persistent or ephemeral. A persistent disk continues to exist and store data even if it is detached from a virtual server or the virtual server to which it is attached shuts down. Ephemeral disks exist and store data only as long as a VM is running. Ephemeral disks store operating system files and other files and data that are deleted when the VM is shut down. Persistent disks are used when you want data to exist on a block storage device independent of a VM. These disks are good options when you have data that you want available independent of the lifecycle of a VM, and support fast operating system- and file system-level access.

Object storage also keeps data independent of the lifecycle of a VM, but it does not support operating system- or file system-level access; you have to use higher-level protocols like HTTP to access objects. It takes longer to retrieve data from object storage than to retrieve it from block storage. You may need a combination of object storage and block storage to meet your application needs. Object storage can store large volumes of data that are copied to persistent disk when needed. This combination gives the advantage of large volumes of storage along with operating system- and file system-based access when needed.

Caches

Caches are in-memory data stores that maintain fast access to data. The time it takes to retrieve data is called *latency*. The latency of in-memory stores is designed to be submillisecond. To give you a comparison, here are some other latencies:

- Making a main memory reference takes 100 nanoseconds, or 0.1 microsecond
- Reading 4KB randomly from an SSD takes 150 microseconds
- Reading 1MB sequentially from memory takes 250 microseconds

- Reading 1MB sequentially from an SSD takes 1,000 microseconds, or 1 millisecond
- Reading 1MB sequentially from disk takes 20,000 microseconds, or 20 milliseconds

Here are some conversions for reference:

- 1,000 nanoseconds equal 1 microsecond.
- 1,000 microseconds equal 1 millisecond.
- 1,000 milliseconds equal 1 second.

These and other useful timing data are available at Jonas Bonér’s “Latency Numbers Every Programmer Should Know” at <https://gist.github.com/jboner/2841832>.

Let’s work through an example of reading 1MB of data. If you have the data stored in an in-memory cache, you can retrieve the data in 250 microseconds, or 0.25 millisecond. If that same data is stored on an SSD, it will take four times as long to retrieve at 1 millisecond. If you retrieve the same data from a hard disk drive, you can expect to wait 20 milliseconds, or 80 times as long as reading from an in-memory cache.

Caches are quite helpful when you need to keep read latency to a minimum in your application. Of course, who doesn’t love fast retrieval times? Why don’t we always store our data in caches? There are three reasons.

- Memory is more expensive than SSD or hard disk drive (HDD) storage. It’s not practical in many cases to have as much in-memory storage as persistent block storage on SSDs or HDDs.
- Caches are volatile; you lose the data stored in the cache when power is lost or the operating system is rebooted. You can store data in a cache for fast access, but it should never be used as the only data store keeping the data. Some form of persistent storage should be used to maintain a “system of truth,” or a data store that always has the latest and most accurate version of the data.
- Caches can get out of synchronization with the system of truth. This can happen if the system of truth is updated but the new data is not written to the cache. When this happens, it can be difficult for an application that depends on the cache to detect the fact that data in the cache is invalid. If you decide to use a cache, be sure to design a cache update strategy that meets your requirements for consistency between the cache and the system of truth. This is such a challenging design problem that it has become memorialized in Phil Karlton’s well-known quip, “There are only two hard things in computer science: cache invalidation and naming things.” (See <https://martinfowler.com/bliki/TwoHardThings.html> for riffs on this rare example of computer science humor.)



Real World Scenario

Improving Database Query Response Time

Users expect web applications to be highly responsive. If a page takes more than 2 to 3 seconds to load, the user experience can suffer. It is common to generate the content of a page using the results of a database query, such as looking up account information

by customer ID. When a query is made to the database, the database engine will look up the data, which is usually on disk. The more users query the database the more queries it has to serve. Databases keep a queue for queries that need to be answered but can't be processed yet because the database is busy with other queries. This can cause longer latency response time, since the web application will have to wait for the database to return the query results.

One way to reduce latency is to reduce the time needed to read the data. In some cases, it helps to replace hard disk drives with faster SSD drives. However, if the volume of queries is high enough that the queue of queries is long even with SSDs, another option is to use a cache.

When query results are fetched, they are stored in the cache. The next time that information is needed, it is fetched from the cache instead of the database. This can reduce latency because data is fetched from memory, which is faster than disk. It also reduces the number of queries to the database, so queries that can't be answered by looking up data in the cache won't have to wait as long in the query queue before being processed.

Networking

When working in the cloud, you'll need to work with networking between your cloud resources and possibly with your on-premise systems.

When you have multiple VMs running in your cloud environment, you will likely need to manage IP addresses at some point. Each network-accessible device or service in your environment will need an IP address. In fact, devices within GCP can have both internal and external addresses. Internal addresses are accessible only to services in your internal GCP network. Your internal GCP network is defined as a virtual private cloud (VPC). External addresses are accessible from the Internet.

External IP addresses can be either static or ephemeral. Static addresses are assigned to a device for extended periods of time. Ephemeral external IP addresses are attached to VMs and released when the VM is stopped.

In addition to specifying IP addresses, you will often need to define firewall rules to control access to subnetworks and VMs in your VPC. For example, you may have a database server that you want to restrict access to so that only an application server can query the database. A firewall rule can be configured to limit inbound and outbound traffic to the IP address of the application server or load balancer in front of the application cluster.

You may need to share data and network access between an on-premise data center and your VPC. You can do this using one of several types of *peering*, which is the general term for linking distinct networks.

Specialized Services

Most public cloud providers offer specialized services that can be used as building blocks of applications or as part of a workflow for processing data. Common characteristics of specialized services are as follows:

- They are serverless; you do not need to configure servers or clusters.
- They provide a specific function, such as translating text or analyzing images.
- They provide an application programming interface (API) to access the functionality of the service.
- As with other cloud services, you are charged based on your use of the service.

These are some of the specialized services in Google Cloud Platform:

- AutoML, a machine learning service
- Cloud Natural Language, a service for analyzing text
- Cloud Vision for analyzing images
- Cloud Inference API, a service for computing correlations over time-series data

Specialized services encapsulate advanced computing capabilities and make them accessible to developers who are not experts in domains, such as natural language processing and machine learning. Expect to see more specialized services added to Google Cloud Platform.

Cloud Computing vs. Data Center Computing

Although it may seem that running VMs in the cloud is not much different from running them in your data center, there are significant differences between operating IT environments in the cloud and an on-premise or colocated data center.

Rent Instead of Own Resources

Corporate data centers are filled with servers, disk arrays, and networking equipment. This equipment is often owned or leased for extended periods by the company, a model that requires companies to either spend a significant amount of money up front to purchase equipment or commit to a long-term lease for the equipment. This approach works well when an organization can accurately predict the number of servers and other equipment it will need for an extended period and it can utilize that equipment consistently.

The model does not work as well when companies have to plan for peak capacity that is significantly higher than the average workload. For example, a retailer may have an average

load that requires a cluster of 20 servers but during the holiday season the workload increases to the point where 80 servers are needed. The company could purchase 80 servers and let 60 idle for most of the year to have resources to accommodate peak capacity. Alternatively, it could purchase or lease fewer servers and tolerate the loss in business that would occur when its compute resources can't keep up with demand. Neither is an appealing option.

Public clouds offer an alternative of short-term rental of compute capacity. The retailer, for example, could run VMs in the cloud during peak periods in addition to its on-premise servers. This gives the retailer access to the servers it needs when it needs them without having to pay for them when they are not needed.

The unit cost of running servers in the cloud may be higher than that of running the equivalent server in the data center, but the total cost of on-premise and short-term in the cloud mix of servers may still be significantly less than the cost of purchasing or leasing for peak capacity and leaving resources idle.

Pay-as-You-Go-for-What-You-Use Model

Related to the short-term rental model of cloud computing is the pay-as-you-go model. When you run a virtual server in the cloud, you will typically pay for a minimum period, such as 10 minutes, and then pay per minute used thereafter. The unit cost per minute will vary depending on the characteristics of the server. Servers with more CPUs and memory will cost more than servers with fewer CPUs and less memory.

It is important for cloud engineers to understand the pricing model of their cloud provider. It is easy to run up a large bill for servers and storage if you are not monitoring your usage. In fact, some cloud customers find that running applications in the cloud can be more expensive than running them on-premise.

Elastic Resource Allocation

Another key differentiator between on-premise and public cloud computing is the ability to add and remove compute and storage resources on short notice. In the cloud, you could start 20 servers in a matter of minutes. In an on-premise data center, it could take days or weeks to do the same thing if additional hardware must be provisioned.

Cloud providers design their data centers with extensive compute, storage, and network resources. They optimize their investment by efficiently renting these resources to customers. With sufficient data about customer use patterns, they can predict the capacity they need to meet customer demand. Since they have many customers, the variation in demand of any one customer has little effect on the overall use of their resources.

Extensive resources and the ability to quickly shift resources between customers enables public cloud providers to offer elastic resource allocation more efficiently than can be done in smaller data centers.

Specialized Services

Specialized services are, by their nature, not widely understood. Many developers understand how to develop user interfaces or query a database, but fewer have been exposed to the details of natural language processing or machine learning. Large enterprises may have the financial resources to develop in-house expertise in areas such as data science and machine vision, but many others don't.

By offering specialized services, cloud providers are bringing advanced capabilities to a wider audience of developers. Like investing in large amounts of hardware, public cloud vendors can invest in specialized services and recover their costs and make a profit because the specialized services are used by a large number of customers.

Exam Essentials

Understand different ways of delivering cloud computing resources. Computing resources can be allocated as individual VMs or clusters of VMs that you manage. You can also use managed kubernetes cluster (GKE) that relieve you of some of the operational overhead of managing a kubernetes cluster. Serverless computing options relieve users of any server management. Instead, developers run their code in a containerized environment managed by the cloud provider or in a compute platform designed for short-running code. Developers and DevOps professionals have the most control over resources when they manage their own servers and clusters. Managed services and serverless options are good choices when you do not need control over the computing environment and will get more value from not having to manage compute resources.

Understand the different forms of cloud storage and when to use them. There are four main categories of storage: object, file, block, and in-memory caches. Object storage is designed for highly reliable and durable storage of objects, such as images or data sets. Object storage has more limited functionality than file system-based storage systems. File system-based storage provides hierarchical directory storage for files and supports common operating system and file system functions. File system services provide network-accessible file systems that can be accessed by multiple servers. Block storage is used for storing data on disks. File systems and databases make use of block storage systems. Block storage is used with persistent storage devices, such as SSDs and HDDs. Caches are in-memory data stores used to minimize the latency of retrieving data. They do not provide persistent storage and should never be considered a “system of truth.”

Understand the differences between running an IT environment on-premise or in the cloud. Running an IT environment in the cloud has several advantages, including short-term rental of resources, pay-as-you-go model, elastic resource allocation, and the ability to use specialized services. The unit cost of cloud resources, such as the cost per minute of a mid-tier server, may be higher in the cloud than on-premise. It is important to understand the cost model of your cloud provider so you can make decisions about the most efficient distribution of workload between cloud and on-premise resources.

Review Questions

1. What is the fundamental unit of computing in cloud computing?
 - A. Physical server
 - B. VM
 - C. Block
 - D. Subnet
2. If you use a cluster that is managed by a cloud provider, which of these will be managed for you by the cloud provider?
 - A. Monitoring
 - B. Networking
 - C. Some security management tasks
 - D. All of the above
3. You need serverless computing for file processing and running the backend of a website; which two products can you choose from Google Cloud Platform?
 - A. Kubernetes Engine and Compute Engine
 - B. App Engine and Cloud Functions
 - C. Cloud Functions and Compute Engine
 - D. Cloud Functions and Kubernetes Engine
4. You have been asked to design a storage system for a web application that allows users to upload large data files to be analyzed by a business intelligence workflow. The files should be stored in a high-availability storage system. File system functionality is not required. Which storage system in Google Cloud Platform should be used?
 - A. Block storage
 - B. Object storage
 - C. Cache
 - D. Network File System
5. All block storage systems use what block size?
 - A. 4KB
 - B. 8KB
 - C. 16KB
 - D. Block size can vary.

6. You have been asked to set up network security in a virtual private cloud. Your company wants to have multiple subnetworks and limit traffic between the subnetworks. Which network security control would you use to control the flow of traffic between subnets?
 - A. Identity access management
 - B. Router
 - C. Firewall
 - D. IP address table
7. When you create a machine learning service to identify text in an image, what type of servers should you choose to manage compute resources?
 - A. VMs
 - B. Clusters of VMs
 - C. No servers; specialized services are serverless
 - D. VMs running Linux only
8. Investing in servers for extended periods of time, such as committing to use servers for three to five years, works well when?
 - A. A company is just starting up
 - B. A company can accurately predict server need for an extended period of time
 - C. A company has a fixed IT budget
 - D. A company has a variable IT budget
9. Your company is based in X and will be running a virtual server for Y. What factor determines the unit per minute cost?
 - A. The time of day the VM is run
 - B. The characteristics of the server
 - C. The application you run
 - D. None of the above
10. You plan to use Cloud Vision to analyze images and extract text seen in the image. You plan to process between 1,000 and 2,500 images per hour. How many VMs should you allocate to meet peak demand?
 - A. 1
 - B. 10
 - C. 25
 - D. None; Cloud Vision is a serverless service.
11. You have to run a number of services to support an application. Which of the following is a good deployment model?
 - A. Run on a large, single VM
 - B. Use containers in a managed cluster

- C. Use two large VMs, making one of them read only
 - D. Use a small VM for all services and increase the size of the VM when CPU utilization exceeds 90 percent
12. You have created a VM. Which of the following system administration operations are you allowed to perform on it?
- A. Configure the file system
 - B. Patch operating system software
 - C. Change file and directory permissions
 - D. All of the above
13. Cloud Filestore is based on what file system technology?
- A. Network File System (NFS)
 - B. XFS
 - C. EXT4
 - D. ReiserFS
14. When setting up a network in GCP, your network the resources in it are treated as what?
- A. Virtual private cloud
 - B. Subdomain
 - C. Cluster
 - D. None of the above
15. You need to store data for X and therefore you are using a cache for Y. How will the cache affect data retrieval?
- A. A cache improves the execution of client-side JavaScript.
 - B. A cache will continue to store data even if power is lost, improving availability.
 - C. Caches can get out of sync with the system of truth.
 - D. Using a cache will reduce latency, since retrieving from a cache is faster than retrieving from SSDs or HDDs.
16. Why can cloud providers offer elastic resource allocation?
- A. Cloud providers can take resources from lower-priority customers and give them to higher-priority customers.
 - B. Extensive resources and the ability to quickly shift resources between customers enables public cloud providers to offer elastic resource allocation more efficiently than can be done in smaller data centers.
 - C. They charge more the more resources you use.
 - D. They don't.

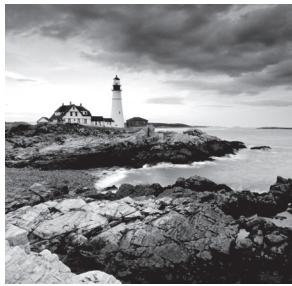
- 17.** What is not a characteristic of specialized services in Google Cloud Platform?
 - A.** They are serverless; you do not need to configure servers or clusters.
 - B.** They provide a specific function, such as translating text or analyzing images.
 - C.** They require monitoring by the user.
 - D.** They provide an API to access the functionality of the service.
- 18.** Your client's transactions must access a drive attached to a VM that allows for random access to parts of files. What kind of storage does the attached drive provide?
 - A.** Object storage
 - B.** Block storage
 - C.** NoSQL storage
 - D.** Only SSD storage
- 19.** You are deploying a new relational database to support a web application. Which type of storage system would you use to store data files of the database?
 - A.** Object storage
 - B.** Data storage
 - C.** Block storage
 - D.** Cache
- 20.** A user prefers services that require minimal setup; why would you recommend Cloud Storage, App Engine, and Cloud Functions?
 - A.** They are charged only by time.
 - B.** They are serverless.
 - C.** They require a user to configure VMs.
 - D.** They can only run applications written in Go.

Chapter 2

Google Cloud Computing Services

THIS CHAPTER COVERS THE FOLLOWING OBJECTIVES OF THE GOOGLE ASSOCIATE CLOUD ENGINEER CERTIFICATION EXAM:

- ✓ 2.2 Planning and configuring compute resources, including selecting appropriate compute choices for a given workload
- ✓ 3.4 Deploying and implementing data solutions, including initializing data systems with products



Google Cloud Platform (GCP) is made up of almost 40 services that meet a variety of computing, storage, and networking needs. This chapter provides an overview of the most important GCP services and describes some important use cases for these services.

Computing Components of Google Cloud Platform

Google Cloud Platform is a suite of cloud computing services that includes compute, storage, and networking services designed to meet the needs of a wide range of cloud computing customers. Small businesses may be attracted to virtual machines (VMs) and storage services. Large businesses and other sizable organizations may be more interested in access to highly scalable clusters of VMs, a variety of relational and NoSQL databases, specialized networking services, and advanced artificial intelligence and machine learning capabilities.

This chapter provides an overview of many of GCP's services. The breadth of services available in the GCP continues to grow. By the time you read this, Google may be offering additional services. Most of the services can be grouped into several core categories.

- Computing resources
- Storage resources
- Databases
- Networking services
- Identity management and security
- Development tools
- Management tools
- Specialized services

A Google-certified Associate Cloud Engineer should be familiar with the services in each category, how they are used, and the advantages and disadvantages of the various services in each category.

Computing Resources

Public cloud services provide a range computing services options. At one end of the spectrum, customers can create and manage VMs themselves. This model gives the cloud user the greatest control of all the computing services. Users can choose the operating system to run, which packages to install, and when to back up and perform other maintenance operations. This type of computing service is typically referred to as infrastructure as a service (IaaS).

An alternative model is called platform as a service (PaaS), which provides a runtime environment to execute applications without the need to manage underlying servers, networks, and storage systems.

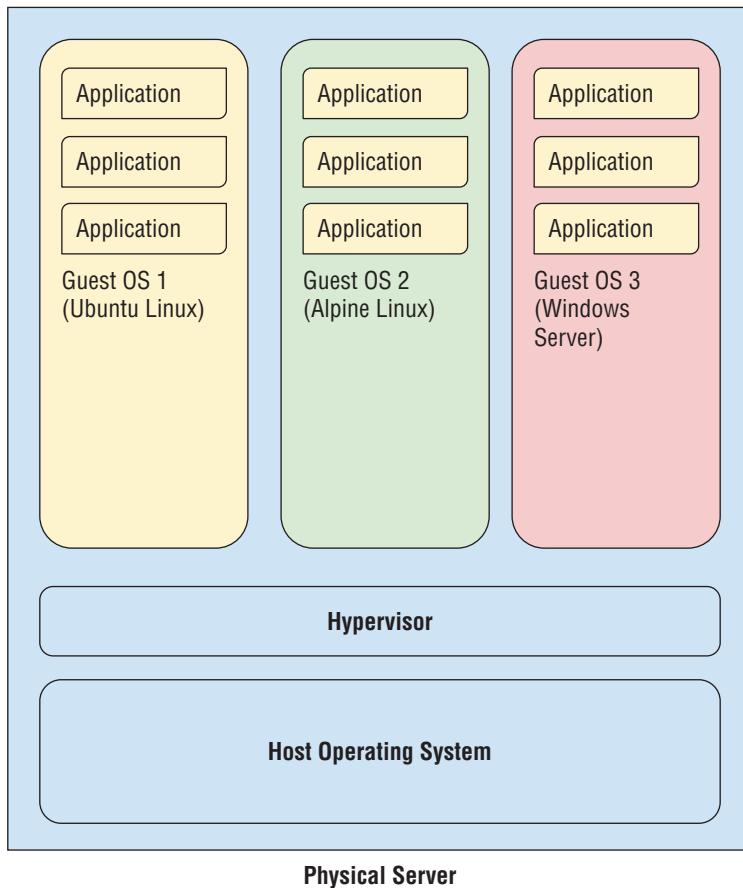
GCP's IaaS computing product is called Compute Engine, and the PaaS offerings are App Engine and Cloud Functions. In addition, Google offers Kubernetes Engine, which is a service for managing containers in a cluster; this type of service is an increasingly popular alternative to managing individual sets of VMs.

Compute Engine

Compute Engine is a service that allows users to create VMs, attach persistent storage to those VMs, and make use of other GCP services, such as Cloud Storage.

VMs are abstractions of physical servers. They are essentially programs that emulate physical servers and provide CPU, memory, storage, and other services that you would find if you ran your favorite operating system on a server under your desk or in a data center. VMs run within a low-level service called a *hypervisor*. GCP uses a security hardened version of the KVM hypervisor. KVM stands for Kernel Virtual Machine and provides virtualization on Linux systems running on x86 hardware.

Hypervisors run on an operating system like Linux or Windows Server. Hypervisors can run multiple operating systems, referred to as *guest operating systems*, while keeping the activities of each isolated from other guest operating systems. Each instance of an executing guest operating system is a VM instance. Figure 2.1 shows the logical organization of VM instances running on a physical server.

FIGURE 2.1 VM instances running within a hypervisor

VMs come in a range of predefined sizes, but you can also create a customized configuration. When you create an instance, you can specify a number of parameters, including the following:

- The operating system
- Size of persistent storage
- Adding graphical processing units (GPUs) for compute-intensive operations like machine learning
- Making the VM preemptible

The last option, making a VM preemptible, means you may be charged significantly less for the VM than normal (around 80 percent less), but your VM could be shut down at any time by Google. It will frequently be shut down if the preemptible VM has run for at least 24 hours.

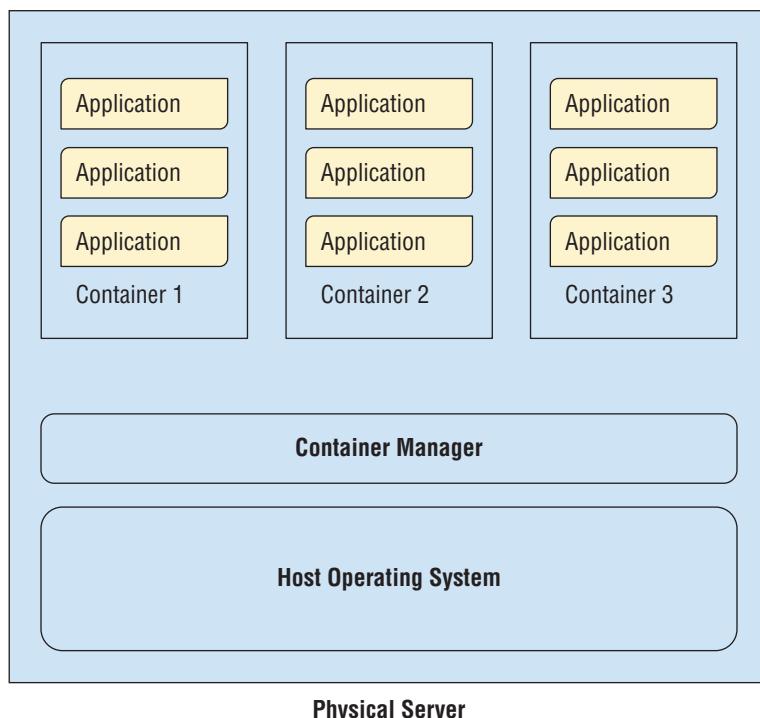
Chapter 4 will introduce the details of managing Compute Engine VMs. To explore Compute Engine, log into the Google Cloud Console, navigate to the main menu on the left, and select Compute Engine.

Kubernetes Engine

Kubernetes Engine is designed to allow users to easily run containerized applications on a cluster of servers. Containers are often compared to VMs because they are each used for isolating computing processing and resources. Containers take a different approach than VMs for isolating computing processes.

As mentioned, a VM runs a guest operating system on a physical server. The physical server runs an operating system as well, along with a hypervisor. Another approach to isolating computing resources is to use features of the host operating system to isolate processes and resources. With this approach, there is no need for a hypervisor; the host operating system maintains isolation. Instead, a container manager is used. That is, a single container manager coordinates containers running on the server. No additional, or guest, operating systems run on top of the container manager. Instead, containers make use of host operating system functionality, while the operating system and container manager ensure isolation between the running containers. Figure 2.2 shows the logical structure of containers.

FIGURE 2.2 Containers running on a physical server



Kubernetes Engine is a GCP product that allows users to describe the compute, storage, and memory resources they'd like to run their services. Kubernetes Engine then provisions the underlying resources. It's easy to add and remove resources from a Kubernetes cluster using a command-line interface or a graphical user interface.

In addition, Kubernetes monitors the health of servers in the cluster and automatically repairs problems, such as failed servers. Kubernetes Engine also supports autoscaling, so if the load on your applications increases, Kubernetes Engine will allocate additional resources.

Chapter 7 will describe the details of planning and managing Kubernetes Engine. To explore Kubernetes Engine, log into the Google Cloud Console, navigate to the main menu on the left, and select Kubernetes Engine.

App Engine

App Engine is GCP's compute PaaS offering. With App Engine, developers and application administrators don't need to concern themselves with configuring VMs or specifying Kubernetes clusters. Instead, developers create applications in a popular programming language such as Java, Go, Python, or Node.js and deploy that code to a serverless application environment.

App Engine manages the underlying computing and network infrastructure. There is no need to configure VMs or harden networks to protect your application. App Engine is well suited for web and mobile backend applications.

App Engine is available in two types: standard and flexible.

In the standard environment, you run applications in a language-specific sandbox, so your application is isolated from the underlying server's operating system as well as from other applications running on that server. The standard environment is well suited to applications that are written in one of the supported languages and do not need operating system packages or other compiled software that would have to be installed along with the application code.

In the flexible environment, you run Docker containers in the App Engine environment. The flexible environment works well in cases where you have application code but also need libraries or other third-party software installed. As the name implies, the flexible environment gives you more options, including the ability to work with background processes and write to local disk.

Chapter 9 will introduce details for using and managing App Engine. To explore App Engine, log into the Google Cloud Console, navigate to the main menu on the left, and select App Engine.

Cloud Functions

Google Cloud Functions is a lightweight computing option that is well suited to event-driven processing. Cloud Functions runs code in response to an event, like a file being uploaded to Cloud Storage or a message being written to a message queue. The code that executes in the Cloud Functions environment must be short-running—this computing service is not designed to execute long-running code. If you need to support long-running applications or jobs, consider Compute Engine, Kubernetes Engine, or App Engine.

Cloud Functions is often used to call other services, such as third-party application programming interfaces (APIs) or other GCP services, like a natural language translation service.

Like App Engine, Cloud Functions is a serverless product. Users only need to supply code; they do not need to configure VMs or create containers. Cloud Functions will automatically scale as load increases.

In addition to the four main computing products, GCP offers a number of storage resources.

Chapter 10 will describe the details of using and managing Cloud Functions. To explore Cloud Functions, log into the Google Cloud Console, navigate to the main menu on the left, and select Cloud Functions.

Storage Components of Google Cloud Platform

Applications and services that run in the cloud have to meet a wide range of requirements when it comes to storage.

Storage Resources

Sometimes an application needs fast read and write times for moderate amounts of data. Other times, a business application may need access to petabytes of archival storage but can tolerate minutes and even hours to retrieve a document. GCP has several storage resources for storing objects and files.

Cloud Storage

Cloud Storage is GCP's object storage system. Objects can be any type of file or binary large object. Objects are organized into buckets, which are analogous to directories in a file system. It is important to remember that Cloud Storage is not a file system. It is a service that receives, stores, and retrieves files or objects from a distributed storage system. Cloud Storage is not part of a VM in the way an attached persistent disk is. Cloud Storage is accessible from VM (or any other network device with appropriate privileges) and so complements file systems on persistent disks.

Each stored object is uniquely addressable by a URL. For example, a .pdf version of this chapter, called *chapter1.pdf*, that is stored in a bucket named *ace-certification-exam-prep* would be addressable as follows:

<https://storage.cloud.google.com/ace-certification-exam-prep/chapter1.pdf>

GCP users and others can be granted permission to read and write objects to a bucket. Often, an application will be granted privileges through IAM roles to enable the application to read and write to buckets.

Cloud Storage is useful for storing objects that are treated as single units of data. For example, an image file is a good candidate for object storage. Images are generally read and written all at once. There is rarely a need to retrieve only a portion of the image. In general, if you write or retrieve an object all at once and you need to store it independently of servers that may or may not be running at any time, then Cloud Storage is a good option.

There are different classes of cloud storage. Regional storage keeps copies of objects in a single Google Cloud *region*. Regions are distinct geographic areas that can have multiple *zones*, or deployment areas. A zone is considered a single failure domain, which means that if all instances of your application are running in a zone and there is a failure, then all instances of your application will be inaccessible. Regional storage is well suited for applications that run in the same region and need low latency access to objects in Cloud Storage.

Cloud Storage has some useful advanced features, such as support for multiple regions. This provides for storing replicas of objects in multiple Google Cloud regions, which is important for high availability, durability, and low latency.



Real World Scenario

Real-World Scenario

If there was an outage in region us-east1 and your objects were stored only in that region, then you would not be able to access those objects during the outage. However, if you enabled multiregion storage, then your objects stored in us-east1 would be stored in another region, such as us-west1, as well.

In addition to high availability and durability, multiregion storage allows for faster access to data when users or applications are distributed across regions.

Sometimes data needs to be kept for extended periods of time but is rarely accessed. In those cases, nearline storage is a good option because it costs less than regional or multiregional storage and is optimized for infrequent access.

The cold storage class is low-cost archival storage designed for high durability and infrequent access. This class of storage is suitable for data that is accessed less than once per year.

A useful feature of Cloud Storage is the set of lifecycle management policies that can automatically manage objects based on policies you define. For example, you could define a policy that moves all objects more than 60 days old in a bucket to nearline storage or deletes any object in a coldline storage bucket that is older than five years.

Persistent Disk

Persistent disks are storage service that are attached to VMs in Compute Engine or Kubernetes Engine. Persistent disks provide block storage on solid-state drives (SSDs) and hard disk drives (HDDs). SSDs are often used for low-latency applications where persistent disk performance is important. SSDs cost more than HDDs, so applications that require

large amounts of persistent disk storage but can tolerate longer read and write times can use HDDs to meet their storage requirements.

An advantage of persistent disks on the Google Cloud Platform is that these disks support multiple readers without a degradation in performance. This allows for multiple instances to read a single copy of data. Disks can also be resized as needed while in use without the need to restart your VMs.

Persistent disks can be up to 64TB in size using either SSDs or HDDs.

Cloud Storage for Firebase

Mobile app developers may find Cloud Storage for Firebase to be the best combination of cloud object storage and the ability to support uploads and downloads from mobile devices with sometimes unreliable network connections.

The Cloud Storage for Firebase API is designed to provide secure transmission as well as robust recovery mechanisms to handle potentially problematic network quality. Once files, like photos or music recordings, are uploaded into Cloud Storage, you can access those files through the Cloud Storage command-line interface and software development kits (SDKs).

Cloud Filestore

Sometimes, developers need to have access to a file system housed on network-attached storage. For these use cases, the Cloud Filestore service provides a shared file system for use with Compute Engine and Kubernetes Engine.

Filestore can provide high numbers of input-output operations per second (IOPS) as well as variable storage capacity. File system administrators can configure Cloud Filestore to meet their specific IOPS and capacity requirements.

Filestore implements the Network File System (NFS) protocol so system administrators can easily mount shared file systems on virtual servers.

Storage systems like the ones just described are used to store coarse-grained objects, like files. When data is more finely structured and has to be retrieved using query languages that describe the subset of data to return, then it is best to use a database management system.

Chapter 11 describes details and guidance for planning storage services. To explore storage options, log into the Google Cloud Console, navigate to the main menu on the left, and select Storage or Filestore.

Databases

GCP provides several database options. Some are relational databases, and some are NoSQL databases. Some are serverless and others require users to manage clusters of servers. Some provide support for atomic transactions, and others are better suited for applications with less stringent consistency and transaction requirements. GCP users must understand their application requirements before choosing a service, and this is especially important when choosing a database, which often provides core storage services in the application stack.

Cloud SQL

Cloud SQL is GCP's managed relational database service that allows users to set up MySQL or PostgreSQL databases on VMs without having to attend to database administration tasks, such as backing up databases or patching database software. Cloud SQL is available in a number of configurations:

- First-generation MySQL databases use MySQL 5.5 or 5.6 and can have up to 16GB of RAM and 500GB of data storage.
- Second-generation MySQL databases use MySQL 5.6 or 5.7 and can have up to 416GB of RAM along with 10TB of data storage. Second-generation MySQL databases can be configured to automatically add storage as needed.
- PostgreSQL 9.6 runs on the second-generation platform and can be configured with up to 64 CPUs, 416GB of RAM, and up to 10TB of storage. Cloud SQL PostgreSQL also supports common extensions such as PostGIS, cubes for analytic processing, and hstore for storing key-value pairs in a single PostgreSQL value.

This database service includes management of replication and allows for automatic failover, providing for highly available databases.

Relational databases are well suited to applications with relatively consistent data structure requirements. For example, a banking database may track account numbers, customer names, addresses, and so on. Since virtually all records in the database will need the same information, this application is a good fit for a relational database.

Cloud Bigtable

Cloud Bigtable is designed for petabyte-scale applications that can manage up to billions of rows and thousands of columns. It is based on a NoSQL model known as a *wide-column data model*, and unlike Cloud SQL that supports relational databases. Bigtable is suited for applications that require low-latency write and read operations. It is designed to support millions of operations per second.

Bigtable integrates with other Google Cloud services, such as Cloud Storage, Cloud Pub/Sub, Cloud Dataflow, and Cloud Dataproc. It also supports the Hbase API, which is an API for data access in the Hadoop big data ecosystem. Bigtable also integrates with open source tools for data processing, graph analysis, and time-series analysis.

Cloud Spanner

Cloud Spanner is Google's globally distributed relational database that combines the key benefits of relational databases, such as strong consistency and transactions, with the ability to scale horizontally like a NoSQL database. Spanner is a high availability database with a 99.999 percent availability Service Level Agreements (SLA), making it a good option for enterprise applications that demand scalable, highly available relational database services.

Cloud Spanner also has enterprise-grade security with encryption at rest and encryption in transit, along with identity-based access controls.

Cloud Spanner supports ANSI 2011 standard SQL.

Cloud Datastore

Cloud Datastore is a NoSQL document database. This kind of database uses the concept of a document, or collection of key-value pairs, as the basic building block. Documents allow for flexible schemas. For example, a document about a book may have key-value pairs listing author, title, and date of publication. Some books may also have information about companion websites and translations into other languages. The set of keys that may be included does not have to be defined prior to use in document databases. This is especially helpful when applications must accommodate a range of attributes, some of which may not be known at design time.

Cloud Datastore is accessed via a REST API that can be used from applications running in Compute Engine, Kubernetes Engine, or App Engine. This database will scale automatically based on load. It will also *shard*, or partition, data as needed to maintain performance. Since Cloud Datastore is a managed service, it takes care of replication, backups, and other database administration tasks.

Although it is a NoSQL database, Cloud Datastore supports transactions, indexes, and SQL-like queries.

Cloud Datastore is well suited to applications that demand high scalability and structured data and do not always need strong consistency when reading data. Product catalogs, user profiles, and user navigation history are examples of the kinds of applications that use Cloud Datastore.

Cloud Memorystore

Cloud Memorystore is an in-memory cache service. Other databases offered in GCP are designed to store large volumes of data and support complex queries, but Cloud Memorystore is a managed Redis service for caching frequently used data in memory. Caches like this are used to reduce the time needed to read data into an application. Cloud Memorystore is designed to provide submillisecond access to data.

As a managed service, Cloud Memorystore allows users to specify the size of a cache while leaving administration tasks to Google. GCP ensures high availability, patching, and automatic failover so users don't have to.

Cloud Firestore

Cloud Firestore is another GCP-managed NoSQL database service designed as a backend for highly scalable web and mobile applications. A distinguishing feature of Cloud Firestore is its client libraries that provide offline support, synchronization, and other features for managing data across mobile devices, IoT devices, and backend data stores. For example, applications on mobile devices can be updated in real time as data in the backend changes.

Cloud Firebase includes a Datastore mode, which enables applications written for Datastore to work with Cloud Firebase as well. When running in Native mode, Cloud Firestore provides real-time data synchronization and offline support.

Cloud Firestore is currently in beta release.

Chapter 12 delves into details of how to create various types of databases, as well as how to load, delete, and query data. Each of the databases can be accessed from the main menu of the Google Cloud Console. From there you can begin to explore how each works and begin to see the differences.

Networking Components of Google Cloud Platform

In this section, we will review the major networking components. Details on setting up networks and managing them are described in Chapters 14 and 15.

Networking Services

Google Cloud Platform provides a number of networking services designed to allow users to configure virtual networks within Google's global network infrastructure, link on-premise data centers to Google's network, optimize content delivery, and protect your cloud resources using network security services.

Virtual Private Cloud

When an enterprise operates its own data center, it controls what is physically located in that data center and connected to its network. Its infrastructure is physically isolated from those of other organizations running in other data centers. When an organization moves to a public cloud, it shares infrastructure with other customers of that public cloud. Although multiple enterprises will use the same cloud infrastructure, each enterprise can logically isolate its cloud resources by creating a virtual private cloud (VPC).

A distinguishing feature of GPC is that a VPC can span the globe without relying on the public Internet. Traffic from any server on a VPC can be securely routed through the Google global network to any other point on that network. Another advantage of the Google network structure is that your backend servers can access Google services, such as machine learning or IoT services, without creating a public IP address for backend servers.

VPCs in the Google Cloud can be linked to on-premise virtual private networks using Internet Protocol Security (IPSec).

Although a VPC is global, enterprises can use separate projects and billing accounts to manage different departments or groups within the organization. Firewalls can be used to restrict access to resources on a VPC as well.

Cloud Load Balancing

Google provides global load balancing to distribute workloads across your cloud infrastructure. Using a single multicast IP address, Cloud Load Balancing can distribute the workload within and across regions, adapt to failed or degraded servers, and autoscale your compute resources to accommodate changes in workload. Cloud Load Balancing also supports internal load balancing, so no IP addresses need to be exposed to the Internet to get the advantages of load balancing.

Cloud Load Balancing is a software service that can load-balance HTTP, HTTPS, TCP/SSL, and UDP traffic.

Cloud Armor

Services exposed to the Internet can become targets of distributed denial-of-service (DDoS) attacks. Cloud Armor is a Google network security service that builds on the Global HTTP(s) Load Balancing service. Cloud Armor features include the following:

- Ability to allow or restrict access based on IP address
- Predefined rules to counter cross-site scripting attacks
- Ability to counter SQL injection attacks
- Ability to define rules at both level 3 (network) and level 7 (application)
- Allows and restricts access based on the geolocation of incoming traffic

Cloud CDN

With content delivery networks (CDNs), users anywhere can request content from systems distributed in various regions. CDNs enable low-latency response to these requests by caching content on a set of endpoints across the globe. Google currently has more than 90 CDN endpoints that are managed as a global resource, so there is no need to maintain region-specific configurations.

CDNs are especially important for sites with large amounts of static content and a global audience. News sites, for example, could use the Cloud CDN service to ensure fast response to requests from any point in the world.

Cloud Interconnect

Cloud Interconnect is a set of GCP services for connecting your existing networks to the Google network. Cloud Interconnect offers two types of connections: interconnects and peering.

Interconnect with direct access to networks uses the Address Allocation for Private Internets standard (RFC 1918) to connect to devices in your VPC. A direct network connection is maintained between an on-premise or hosted data center and one of Google's colocation facilities, which are located in North America, South America, Europe, Asia,

and Australia. Alternatively, if an organization cannot achieve a direct interconnect with a Google facility, it could use Partner Interconnect. This service depends on a third-party network provider to provide connectivity between the company's data center and a Google facility.

For organizations that do not require the bandwidth of a direct or peered interconnect, Google offers VPN services that enable traffic to transmit between data centers and Google facilities using the public Internet.

Cloud DNS

Cloud DNS is a domain name service provided in GCP. Cloud DNS is a high availability, low-latency service for mapping from domain names, such as `example.com`, to IP addresses, such as 74.120.28.18.

Cloud DNS is designed to automatically scale so customers can have thousands and millions of addresses without concern for scaling the underlying infrastructure. Cloud DNS also provides for private zones that allow you to create custom names for your VMs if you need those.

Identity Management

GCP's Cloud Identity and Access Management (IAM) service enables customers to define fine-grained access controls on resources in the cloud. IAM uses the concepts of users, roles, and privileges.

Identities are abstractions about users of services, such as a human user. After an identity is authenticated by logging in or some other mechanism, the authenticated user can access resources and perform operations based on the privileges granted to that identity. For example, a user may have the privilege to create a bucket in Cloud Storage or delete a VM running in Compute Engine.

Users often need similar sets of permissions. Someone who has the ability to create a VM will likely want to be able to modify or delete those VMs. Groups of related permissions can be bundled into roles. Roles are sets of permissions that can be assigned to an identity.

As a Google Certified Associate Cloud Engineer, you will become familiar with identities, roles, and permissions and how to administer them across organizations and projects.

You can find identity management tools under the IAM and admin menu in the Google Cloud Console. Chapter 17 provides details on identity, roles, and best practices for their management.

Development Tools

Google Cloud Platform is an excellent choice for developers and software engineers because of the easy access to infrastructure and data management services, but also for the tools it supports.

Cloud SDK is a command-line interface for managing GCP resources, including VMs, disk storage, network firewalls, and virtually any other resource you might deploy in GCP. In addition to a command-line interface, Cloud SDK has client libraries for Java, Python, Node.js, Ruby, GO, .NET, and PHP.

GCP also supports deploying applications to containers with Container Registry, Cloud Build, and Cloud Source Repositories.

Google has also developed plug-ins to make it easy to work with popular development tools. These include the following:

- Cloud Tools for IntelliJ
- Cloud Tools for PowerShell
- Cloud Tools for Visual Studio
- Cloud Tools for Eclipse
- App Engine Gradle Plugin
- App Engine Maven Plugin

Of course, applications move from development to production deployment, and GCP follows that flow with additional management tools to help monitor and maintain applications after they are deployed.

Additional Components of Google Cloud Platform

Management tools are designed for DevOps professionals who are responsible for ensuring the reliability, availability, and scalability of applications.

Management Tools

The following are some of the most important tools in the management tools category:

Stackdriver This is a service that collects metrics, logs, and event data from applications and infrastructure and integrates the data so DevOps engineers can monitor, assess, and diagnose operational problems.

Monitoring This extends the capabilities of Stackdriver by collecting performance data from GCP, AWS resources, and application instrumentation, including popular open source systems like NGINX, Cassandra, and Elasticsearch.

Logging This service enables users to store and analyze and alert on log data from both GCP and AWS logs.

Error Reporting This aggregates application crash information for display in a centralized interface.

Trace This is a distributed tracing service that captures latency data about an application to help identify performance problem areas.

Debugger This enables developers to inspect the state of executing code, inject commands, and view call stack variables.

Profiler This is used to collect CPU and memory utilization information across the call hierarchy of an application. Profiler uses statistical sampling to minimize the impact of profiling on application performance.

The combination of management tools provides insights into applications as they run in production, enabling more effective monitoring and analysis of operational systems.

Specialized Services

In addition to IaaS and PaaS offerings, GCP has specialized services for APIs, data analytics, and machine learning.

Apigee API Platform

The Apigee API platform is a management service for GCP customers providing API access to their applications. The Apigee platform allows developers to deploy, monitor, and secure their APIs. It also generates API proxies based on the Open API Specification.

It is difficult to predict load on an API, and sometimes spikes in use can occur. For those times, the Apigee API platform provides routing and rate-limiting based on policies customers can define.

APIs can be authenticated using either OAuth 2.0 or SAML. Data is encrypted both in transit and at rest in the Apigee API platform.

Data Analytics

GCP has a number of services designed for analyzing big data in batch and streaming modes. Some of the most important tools in this set of services include the following:

- BigQuery, a petabyte-scale analytics database service for data warehousing
- Cloud Dataflow, a framework for defining batch and stream processing pipelines
- Cloud Dataproc, a managed Hadoop and Spark service
- Cloud Dataprep, a service that allows analysts to explore and prepare data for analysis

Often, data analytics and data warehousing projects use several of these services together.

AI and Machine Learning

Google is a leader in AI and machine learning, so it is no surprise that GCP includes several AI services. Specialized services in this area include the following:

Cloud AutoML This is a tool that allows developers without machine learning experience to develop machine learning models.

Cloud Machine Learning Engine This is a platform for building and deploying scalable machine learning systems to production.

Cloud Natural Language Processing This tool is for analyzing human languages and extracting information from text.

Cloud Vision This is an image analysis platform for annotating images with metadata, extracting text, or filtering content.

Exam Essentials

Understand the differences between Compute Engine, Kubernetes Engine, App Engine, and Cloud Functions. Compute Engine is Google’s VM service. Users can choose CPUs, memory, persistent disks, and operating systems. They can further customize a VM by adding graphics processing units for compute-intensive operations. VMs are managed individually or in groups of similar servers.

Kubernetes Engine manages groups of virtual servers and applications that run in containers. Containers are lighter weight than VMs. Kubernetes is called an *orchestration service* because it distributes containers across clusters, monitors cluster health, and scales as prescribed by configurations.

App Engine is Google’s PaaS. Developers can run their code in a language-specific sandbox when using the standard environment or in a container when using the flexible environment. App Engine is a serverless service, so customers do not need to specify VM configurations or manage servers.

Cloud Functions is a serverless service that is designed to execute short-running code that responds to events, such as file uploads or messages being published to a message queue. Functions may be written in Node.js or Python.

Understand what is meant by serverless. Serverless means customers using a service do not need to configure, monitor, or maintain the computing resources underlying the service. It does not mean there are no servers involved—there are always physical servers that run applications, functions, and other software. Serverless only refers to not needing to manage those underlying resources.

Understand the difference between object and file storage. Object stores are used to store and access file-based resources. These objects are referenced by a unique identifier, such as a URL. Object stores do not provide block or file system services, so they are not suitable for database storage. Cloud Storage is GCP's object storage service.

File storage supports block-based access to files. Files are organized into directories and subdirectories. Google's Filestore is based on the NFS.

Know the different kinds of databases. Databases are broadly divided into relational and NoSQL databases.

Relational databases support transactions, strong consistency, and the SQL query languages. Relational databases have been traditionally difficult to horizontally scale. Cloud Spanner is a global relational database that provides the advantages of relational databases with the scalability previously found only in NoSQL databases.

NoSQL databases are designed to be horizontally scalable. Other features, such as strong consistency and support for standard SQL, are often sacrificed to achieve scalability and low-latency query responses. NoSQL databases may be key-value stores like Cloud Memorystore, document databases like Cloud Datastore, or wide-column databases such as Cloud Bigtable.

Understand virtual private clouds. A VPC is a logical isolation of an organization's cloud resources within a public cloud. In GCP, VPCs are global; they are not restricted to a single zone or region. All traffic between GCP services can be transmitted over the Google network without the need to send traffic over the public Internet.

Understand load balancing. Load balancing is the process of distributing a workload across a group of servers. Load balancers can route workload based on network-level or application-level rules. GCP load balancers can distribute workloads globally.

Understand developer and management tools. Developer tools support common workflows in software engineering, including using version control for software, building containers to run applications and services, and making containers available to other developers and orchestration systems, such as Kubernetes Engine.

Management tools, such as Stackdriver, Monitoring, and Logging, are designed to provide systems administration information to developers and operators who are responsible for ensuring applications are available and operating as expected.

Know the types of specialized services offered by Google Cloud Platform. GCP includes a growing list of specialized services for data analytics, and AI and machine learning.

Know the main differences between on-premises and public cloud computing. On-premise computing is computing, storage, networking, and related services that occur on infrastructure managed by a company or organization for its own use. Hardware may be located literally on the premises in a company building or in a third-party colocation facility. Colocation facilities provide power, cooling, and physical security, but the customers of the colocation facility are responsible for all the setup and management of the infrastructure.

Public cloud computing uses infrastructure and services provided by a cloud provider such as Google, AWS, or Microsoft. The cloud provider maintains all physical hardware and facilities. It provides a mix of services, such as VMs that are configured and maintained by customers and serverless offerings that enable customers to focus on application development-while the cloud provider takes on more responsibility for maintaining the underlying compute infrastructure.

Review Questions

You can find the answers in the Appendix.

1. You are planning to deploy a SaaS application for customers in North America, Europe, and Asia. To maintain scalability, you will need to distribute workload across servers in multiple regions. Which GCP service would you use to implement the workload distribution?
 - A. Cloud DNS
 - B. Cloud Spanner
 - C. Cloud Load Balancing
 - D. Cloud CDN
2. You have decided to deploy a set of microservices using containers. You could install and manage Docker on Compute Engine instances, but you'd rather have GCP provide some container management services. Which two GCP services allow you to run containers in a managed service?
 - A. App Engine standard environment and App Engine flexible environment
 - B. Kubernetes Engine and App Engine standard environment
 - C. Kubernetes Engine and App Engine flexible environment
 - D. App Engine standard environment and Cloud Functions
3. Why would an API developer want to use the Apigee API platform?
 - A. To get the benefits of routing and rate-limiting
 - B. Authentication services
 - C. Version control of code
 - D. A and B
 - E. All of the above
4. You are deploying an API to the public Internet and are concerned that your service will be subject to DDoS attacks. Which GCP service should you consider to protect your API?
 - A. Cloud Armor
 - B. Cloud CDN
 - C. Cloud IAM
 - D. VPCs

5. You have an application that uses a Pub/Sub message queue to maintain a list of tasks that are to be processed by another application. The application that consumes messages from the Pub/Sub queue removes the message only after completing the task. It takes approximately 10 seconds to complete a task. It is not a problem if two or more VMs perform the same task. What is a cost-effective configuration for processing this workload?
 - A. Use preemptible VMs
 - B. Use standard VMs
 - C. Use DataProc
 - D. Use Spanner
6. Your department is deploying an application that has a database backend. You are concerned about the read load on the database server and want to have data available in memory to reduce the time to respond to queries and to reduce the load on the database server. Which GCP service would you use to keep data in memory?
 - A. Cloud SQL
 - B. Cloud Memorystore
 - C. Cloud Spanner
 - D. Cloud Datastore
7. The Cloud SDK can be used to configure and manage resources in which of the following services?
 - A. Compute Engine
 - B. Cloud Storage
 - C. Network firewalls
 - D. All of the above
8. What server configuration is required to use Cloud Functions?
 - A. VM configuration
 - B. Cluster configuration
 - C. Pub/Sub configuration
 - D. None
9. You have been assigned the task of consolidating log data generated by each instance of an application. Which of the Stackdriver management tools would you use?
 - A. Monitoring
 - B. Trace
 - C. Debugger
 - D. Logging

- 10.** Which specialized services are most likely to be used to build a data warehousing platform that requires complex extraction, transformation, and loading operations on batch data as well as processing streaming data?
- A.** Apigee API platform
 - B.** Data analytics
 - C.** AI and machine learning
 - D.** Cloud SDK
- 11.** Your company has deployed 100,000 Internet of Things (IoT) sensors to collect data on the state of equipment in several factories. Each sensor will collect and send data to a data store every 5 seconds. Sensors will run continuously. Daily reports will produce data on the maximum, minimum, and average value for each metric collected on each sensor. There is no need to support transactions in this application. Which database product would you recommend?
- A.** Cloud Spanner
 - B.** Cloud Bigtable
 - C.** Cloud SQL MySQL
 - D.** Cloud SQL PostgreSQL
- 12.** You are the lead developer on a medical application that uses patients' smartphones to capture biometric data. The app is required to collect data and store it on the smartphone when data cannot be reliably transmitted to the backend application. You want to minimize the amount of development you have to do to keep data synchronized between smartphones and backend data stores. Which data store option should you recommend?
- A.** Cloud Firestore
 - B.** Cloud Spanner
 - C.** Cloud Datastore
 - D.** Cloud SQL
- 13.** A software engineer comes to you for a recommendation. She has implemented a machine learning algorithm to identify cancerous cells in medical images. The algorithm is computationally intensive, makes many mathematical calculations, requires immediate access to large amounts of data, and cannot be easily distributed over multiple servers. What kind of Compute Engine configuration would you recommend?
- A.** High memory, high CPU
 - B.** High memory, high CPU, GPU
 - C.** Mid-level memory, high CPU
 - D.** High CPU, GPU

- 14.** You are tasked with mapping the authentication and authorization policies of your on-premises applications to GPC's authentication and authorization mechanisms. The GCP documentation states that an identity must be authenticated in order to grant privileges to that identity. What does the term *identity* refer to?
- A.** VM ID
 - B.** User
 - C.** Role
 - D.** Set of privileges
- 15.** A client is developing an application that will need to analyze large volumes of text information. The client is not expert in text mining or working with language. What GCP service would you recommend they use?
- A.** Cloud Vision
 - B.** Cloud ML
 - C.** Cloud Natural Language Processing
 - D.** Cloud Text Miner
- 16.** Data scientists in your company want to use a machine learning library available only in Apache Spark. They want to minimize the amount of administration and DevOps work. How would you recommend they proceed?
- A.** Use Cloud Spark
 - B.** Use Cloud Dataproc
 - C.** Use Bigquery
 - D.** Install Apache Spark on a cluster of VMs
- 17.** Database designers at your company are debating the best way to move a database to GCP. The database supports an application with a global user base. Users expect support for transactions and the ability to query data using commonly used query tools. The database designers decide that any database service they choose will need to support ANSI 2011 and global transactions. Which database service would you recommend?
- A.** Cloud SQL
 - B.** Cloud Spanner
 - C.** Cloud Datastore
 - D.** Cloud Bigtable
- 18.** Which specialized service supports both batch and stream processing workflows?
- A.** Cloud Dataproc
 - B.** Bigquery
 - C.** Cloud Datastore
 - D.** AutoML

- 19.** You have a Python application you'd like to run in a scalable environment with the least amount of management overhead. Which GCP product would you select?
- A. App Engine flexible environment
 - B. Cloud Engine
 - C. App Engine standard environment
 - D. Kubernetes Engine
- 20.** A product manager at your company reports that customers are complaining about the reliability of one of your applications. The application is crashing periodically, but developers have not found a common pattern that triggers the crashes. They are concerned that they do not have good insight into the behavior of the application and want to perform a detailed review of all crash data. Which Stackdriver tool would you use to view consolidated crash information?
- A. DataProc
 - B. Monitoring
 - C. Logging
 - D. Error Reporting

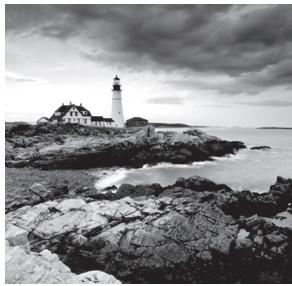
Chapter 3



Projects, Service Accounts, and Billing

THIS CHAPTER COVERS THE FOLLOWING OBJECTIVES OF THE GOOGLE ASSOCIATE CLOUD ENGINEER CERTIFICATION EXAM:

- ✓ 1.1 Setting up cloud projects and accounts
- ✓ 1.2 Managing billing configuration



Before delving into computing, storage, and networking services, we need to discuss how Google Cloud Platform (GCP) organizes resources and links the use of those resources to a

billing system. This chapter introduces the GCP organizational hierarchy, which consists of organizations, folders, and projects. It also discusses service accounts, which are ways of assigning roles to compute resources so they can carry out functions on your behalf. Finally, the chapter briefly discusses billing.

How GCP Organizes Projects and Accounts

When you use GCP, you probably launch virtual machines or clusters, maybe create buckets to storage objects, and make use of serverless computing services such as App Engine and Cloud Functions. The list of resources you use can grow quickly. They can also change in dynamic, unpredictable ways as autoscaling services respond to workload.

If you run a single application or a few services for your department, you might be able to track all resources by viewing lists of resources in use. As the scope of your GCP use grows, you will probably have multiple departments, each with its own administrators who need different privileges. GCP provides a way to group resources and manage them as a single unit. This is called the *resource hierarchy*. The access to resources in the resource hierarchy is controlled by a set of policies that you can define.

GCP Resource Hierarchy

The central abstraction for managing GCP resources is the resource hierarchy. It consists of three levels:

- Organization
- Folder
- Project

Let's describe how these three components relate to each other.

Organization

An organization is the root of the resource hierarchy and typically corresponds to a company or organization. G-suite domains and a Cloud Identity accounts map to GCP organizations. G Suite is Google's office productivity suite, which includes Gmail, Docs, Drive, Calendar, and other services. If your company uses G Suite, you can create an organization in your GCP hierarchy. If your company does not use G Suite, you can use Cloud Identity, Google's identity as a service (IDaaS) offering (Figure 3.1).

FIGURE 3.1 You can create Cloud Identity accounts and manage G Suite users from the Identity & Organization form.

The screenshot shows the Google Cloud IAM & admin interface. On the left, a sidebar lists various IAM-related options: IAM, Identity & Organization (which is selected and highlighted in blue), Organization policies, Quotas, Service accounts, Labels, Privacy & Security, Settings, Cryptographic keys, Identity-Aware Proxy, Roles, and Audit Logs. The main content area is titled "Identity & Organization BETA". It displays a "Signup completed" message with the date "Oct 1, 2018" and the URL "dsgcpcert.com". Below this, there is a "Migrate projects and billing accounts" section with a "REQUEST" and "ACCEPT" button. Further down, there is a "Set permissions" section with a "SET PERMISSIONS" button. A "How to set up organizations" section follows, with a "DELEGATE SETUP" button. At the bottom, there is an "Admin Console" section with a link to "admin.google.com" and "MANAGE USERS" and "MANAGE GROUPS" buttons.

A single cloud identity is associated with at most one organization. Cloud identities have super admins, and those super admins assign the role of Organization Administrator Identity and Access Management (IAM) role to users who manage the organization. In addition, GCP will automatically grant Project Creator and Billing Account Creator IAM roles to all users in the domain. This allows any user to create projects and enable billing for the cost of resources used.

The users with the Organization Administrator IAM role are responsible for the following:

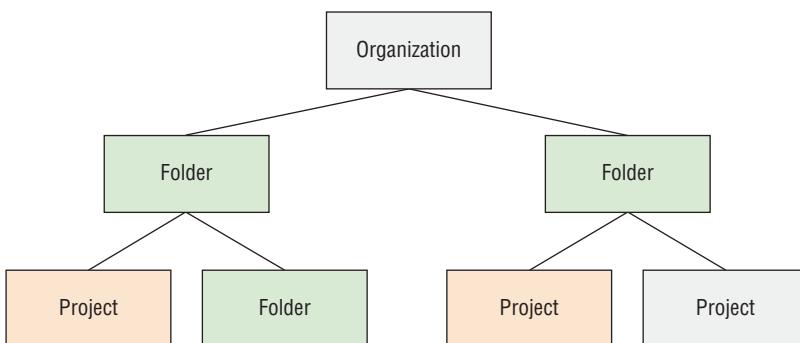
- Defining the structure of the resource hierarchy
- Defining identity access management policies over the resource hierarchy
- Delegating other management roles to other users

When a member of a G Suite organization / Cloud Identity account creates a billing account or project, GCP will automatically create an organization resource. All projects and billing accounts will be children of the organization resource. In addition, when the organization is created, all users in that organization are granted Project Creator and Billing Account Creator roles. From that point on, G Suite users will have access to GCP resources.

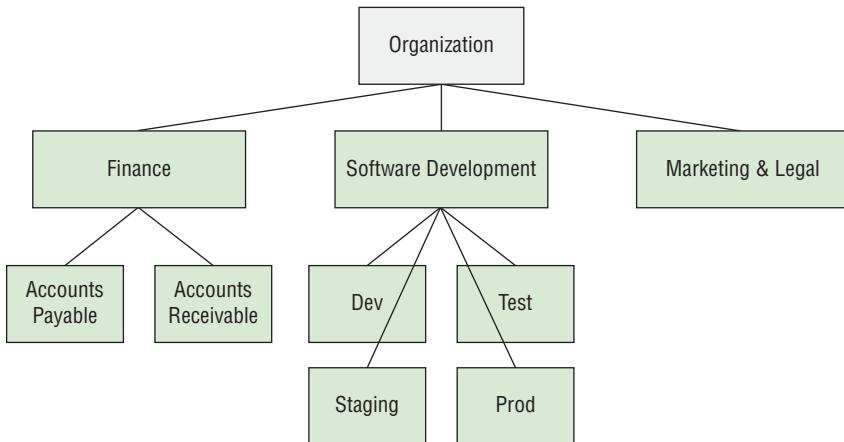
Folder

Folders are the building blocks of multilayer organizational hierarchies. Organizations contain folders. Folders can contain other folders or projects. A single folder may contain both folders and projects (see Figure 3.2). Folder organization is usually built around the kinds of services provided by resources in the contained projects and the policies governing folders and projects.

FIGURE 3.2 Generic organization folder project



Consider an example resource hierarchy. An organization has four departments: finance, marketing, software development, and legal. The finance department has to keep its accounts receivable and accounts payable resources separate, so the administrator creates two folders within the Finance folder: Accounts Receivable and Accounts Payable. Software development uses multiple environments, including Dev, Test, Staging, and Production. Access to each of the environments is controlled by policies specific to that environment, so it makes sense to organize each environment into its own folder. Marketing and legal can have all of their resources shared across members of the department, so a single folder is sufficient for both of those departments. Figure 3.3 shows the organization hierarchy for this organization.

FIGURE 3.3 Example organization folder project

Now that we have an organization defined and have set up folders that correspond to our departments and how different groups of resources will be accessed, we can create projects.

Project

Projects are in some ways the most important part of the hierarchy. It is in projects that we create resources, use GCP services, manage permissions, and manage billing options.

The first step in working with a project is to create one. Anyone with the `resourcemanager.projects.create` IAM permission can create a project. By default, when an organization is created, every user in the domain is granted that permission.

Your organization will have a quota of projects it can create. The quota can vary between organizations. Google makes decisions about project quotas based on typical use, the customer's usage history, and other factors. If you reach your limit of projects and try to create another, you will be prompted to request an increase in the quota. You'll have to provide information such as the number of additional projects you need and what they will be used for.

After you have created your resource hierarchy, you can define policies that govern it.

Organization Policies

GCP provides an Organization Policy Service. This service controls access to an organization's resources. The Organization Policy Service complements the IAM service. The IAM lets you assign permissions so users or roles can perform specific operations in the cloud. The Organization Policy Service lets you specify limits on the ways resources can be

used. One way to think of the difference is that IAM specifies who can do things, and the Organization Policy Service specifies what can be done with resources.

The organization policies are defined in terms of constraints on a resource.

Constraints on Resources

Constraints are restrictions on services. GCP has list constraints and Boolean constraints.

List constraints are lists of values that are allowed or disallowed for a resource. The following are some types of list constraints:

- Allow a specific set of values
- Deny a specific set of values
- Deny a value and all its child values
- Allow all allowed values
- Deny all values

Boolean constraints evaluate to true or false and determine whether the constraint is applied or not. For example, if you want to deny access to serial ports on VMs, you can set constraints/compute.disableSerialPortAccess to TRUE.

Policy Evaluation

Organizations may have standing policies to protect data and resources in the cloud. For example, there may be rules dictating who in the organization can enable a service API or create a service account. Your InfoSec department may require that all VMs disable serial port access. You could implement controls on each individual VM, but that is inefficient and prone to error. A better approach is to define a policy that constrains what can be done and attach that policy to an object in the resource hierarchy.

For example, since InfoSec wants all VMs to disable serial port access, you could specify a policy that constrains serial port access and then attach it to the organization. All folders and projects below the organization will inherit that policy. Since policies are inherited and cannot be disabled or overridden by objects lower in the hierarchy, this is an effective way to apply a policy across all organizational resources.

Policies are managed through the Organization Policies form in the IAM & admin form. Figure 3.4 shows an example set of policies.

Multiple policies can be in effect for a folder or project. For example, if the organization had a policy on serial port access and a folder containing a project had a policy limiting who can create service accounts, then the project would inherit both policies and both would constrain what could be done with resources in that project.

FIGURE 3.4 Organizational policies are managed in the IAM & admin console.

The screenshot shows the 'Organization policies' page under the 'IAM & admin' section. On the left, a sidebar lists various administrative options: IAM, Identity & Organization, Organization policies (which is selected and highlighted in blue), Quotas, Service accounts, Labels, Privacy & Security, Settings, Cryptographic keys, Identity-Aware Proxy, Roles, and Audit Logs. The main content area displays a table of organization policies. The table has two columns: 'Name' and 'ID'. The policies listed are:

Name	ID
Compute Storage resource use restrictions (Compute Engine disks, images, and snapshots)	constraints/compute.storageResourceUseRestrictions
Define allowed APIs and services	constraints/serviceuser.services
Define allowed external IPs for VM instances	constraints/compute.vmExternalIpAccess
Define trusted image projects	constraints/compute.trustedImageProjects
Disable Guest Attributes Compute Engine Metadata	constraints/compute.disableGuestAttributesAccess
Disable service account creation	constraints/iam.disableServiceAccountCreation
Disable service account key creation	constraints/iam.disableServiceAccountKeyCreation
Disable VM nested virtualization	constraints/compute.disableNestedVirtualization
Disable VM serial port access	constraints/compute.disableSerialPortAccess
Domain restricted sharing	constraints/iam.allowedPolicyMemberDomains
Google Cloud Storage - retention policy duration in seconds	constraints/storage.retentionPolicySeconds
Restrict shared VPC project lien removal	constraints/compute.restrictXpnProjectLienRemoval

Managing Projects

One of the first tasks you will perform when starting a new cloud initiative is to set up a project. This can be done with the Google Cloud Console. Assuming you have created an account with GCP, navigate to the Google Cloud Console at <https://console.cloud.google.com> and log in. You will see the home page, which looks something like Figure 3.5.

FIGURE 3.5 Home page console

The screenshot shows the Google Cloud Platform home page for the project 'My First Project'. The top navigation bar includes links for 'Google Cloud Platform', 'My First Project', a search bar, and user profile icons. Below the navigation is a 'DASHBOARD' tab and an 'ACTIVITY' section. The main content area is divided into several cards:

- Project info**: Shows the project name ('My First Project'), project ID ('second-grail-218201'), and project number ('676062005948'). It also has a link to 'Go to project settings'.
- APIs**: A chart titled 'Requests (requests/sec)' showing data for the time frame from 10 AM to 10:45. The chart indicates 'No data is available for the selected time frame.' It has a link to 'Go to APIs overview'.
- Google Cloud Platform status**: Shows 'All services normal' and a link to 'Go to Cloud status dashboard'.
- Error Reporting**: Shows 'No sign of any errors. Have you set up Error Reporting?' and a link to 'Learn how to set up Error Reporting'.
- News**: Displays recent news items:
 - Last month today: September on GCP (1 hour ago)
 - A developer onramp to Kubernetes with GKE (17 hours ago)
 - Google Cloud Platform: Your cloud destination for mission critical SAP workloads (1 day ago)
- Resources**: Shows 'This project has no resources'.
- Trace**: Shows 'No trace data from the past 7 days'.

From the Navigation menu in the upper-left corner, select IAM & admin and then select Manage Resources (see Figure 3.6 and Figure 3.7).

FIGURE 3.6 Navigation menu

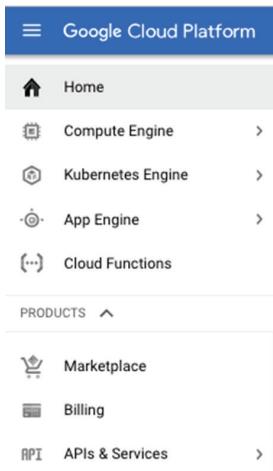
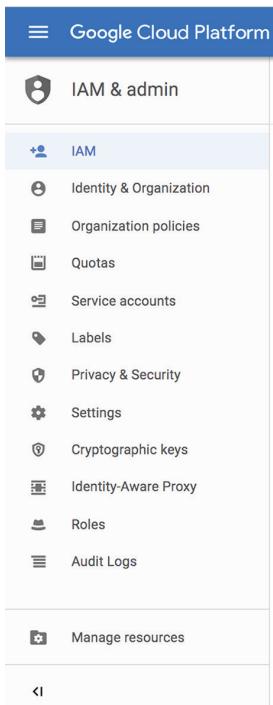


FIGURE 3.7 Select Manage Resources.



From there, you can click Create Project, which displays the Create Project dialog. You can enter the name of a project and select an organization in this dialog (Figure 3.8 and Figure 3.9).

FIGURE 3.8 Click Create Project.

The screenshot shows the Google Cloud Platform dashboard. At the top, there's a blue header bar with the text "Google Cloud Platform". Below it, a navigation bar includes a back arrow, the text "Manage Sources", a "+ CREATE PROJECT" button, a search icon, a bell icon, and a user profile icon. A "SHOW INFO PANEL" link is also present. The main content area has a title "Manage Sources" and a sub-section "Create Project". There's a search bar labeled "Filter by name, ID, project number, or label" and a "Columns" dropdown. A table lists one project: "My First Project" with "Project ID" "second-grail-218201".

FIGURE 3.9 Create Project dialog

The screenshot shows the "New Project" dialog. At the top, there's a blue header bar with the text "Google Cloud Platform". Below it, the title "New Project" is displayed. The dialog contains a warning message: "⚠ You have 24 projects remaining in your quota. Request an increase or delete projects." with a "Learn more" link and a "MANAGE QUOTAS" button. Below this, there's a "Project Name" input field with a red asterisk indicating it's required. A note below the field says "Your project ID will be based on your project name. It cannot be changed later." To the right of the input field is a help icon (?). Below the input field is a "Location" section with a "No organization" dropdown and a "BROWSE" button. A note below the location section says "Parent organization or folder". At the bottom of the dialog are two buttons: a blue "CREATE" button and a white "CANCEL" button.

Note that when you create a project, your remaining quota of projects is displayed. If you need additional projects, click the Manage Quotas link to request an increase in your quota.

Roles and Identities

In addition to managing resources, as a cloud engineer you will have to manage access to those resources. This is done with the use of roles and identities.

Roles in GCP

A *role* is a collection of permissions. Roles are granted to users by binding a user to a role. When we talk of identities, we mean the record we use to represent a human user or service

account in GCP. For example, Alice is a software engineer developing applications in the cloud (the human user), and she has an identity with a name such as `alice@example.com`. Roles are assigned to `alice@example.com` within GCP so that Alice can create, modify, delete, and use resources in GCP.

There are three types of roles in Google Cloud Platform:

- Primitive roles
- Predefined roles
- Custom roles

Primitive roles include Owner, Editor, and Viewer. These are basic privileges that can be applied to most resources. It is a best practice to use predefined roles instead of primitive roles when possible. Primitive roles grant wide ranges of permissions that may not always be needed by a user. By using predefined roles, you can grant only the permissions a user needs to perform their function. This practice of only assigning permissions that are needed and no more is known as the *principle of least privilege*. It is one of the fundamental best practices in information security.

Predefined roles provide granular access to resources in GCP, and they are specific to GCP products. (See Figure 3.10.) For example, App Engine roles include the following:

- `appengine.appAdmin`, which grants identities the ability to read, write, and modify all application settings
- `appengine.ServiceAdmin`, which grants read-only access to application settings and write-level access to module-level and version-level settings
- `appengine.appViewer`, which grants read-only access to applications.

FIGURE 3.10 A sample list of roles in GCP

Roles for "dsgcpcert.com" organization

A role is a group of permissions that you can assign to members. You can create a role and add permissions to it, or copy an existing role and adjust its permissions. [Learn more](#)

Type	Title	Used in	Status
Access Context Manager Admin	Access Context Manager Admin	Other	Enabled
Access Context Manager Editor	Access Context Manager Editor	Other	Enabled
Access Context Manager Reader	Access Context Manager Reader	Other	Enabled
Access Transparency Admin	Access Transparency Admin	Organization Policy	Enabled
Admin	Admin	Cloud Talent Solution	Enabled
Admin of Tenancy Units	Admin of Tenancy Units	Service Consumer Management	Enabled
Android Management User	Android Management User	Android Management	Enabled
API Keys Admin	API Keys Admin	Service Usage	Enabled
API Keys Viewer	API Keys Viewer	Service Usage	Enabled
App Engine Admin	App Engine Admin	App Engine	Enabled
App Engine Code Viewer	App Engine Code Viewer	App Engine	Enabled
App Engine Deployer	App Engine Deployer	App Engine	Enabled
App Engine Service Admin	App Engine Service Admin	App Engine	Enabled
App Engine Viewer	App Engine Viewer	App Engine	Enabled

Custom roles allow cloud administrators to create and administer their own roles. Custom roles are assembled using permissions defined in IAM. While you can use most permissions in a custom role, some, such as `iam.ServiceAccounts.getAccessToken`, are not available in custom roles.

Granting Roles to Identities

Once you have determined which roles you want to provide to users, you can assign roles to users through the IAM console. It is important to know that permissions cannot be assigned to users. They can be assigned only to roles. Roles are then assigned to users.

From the IAM console, you can select a project that will display a permission interface, such as in Figure 3.11.

FIGURE 3.11 IAM permissions

The screenshot shows the Google Cloud Platform IAM & admin interface. On the left, there's a sidebar with options like IAM, Identity & Organization, Organization policies, Quotas, Service accounts, Labels, and Privacy & Security. The main area is titled 'Permissions for project "My First Project"'. It says 'These permissions affect this project and all of its resources.' and has a 'Learn more' link. Below that, it says 'View By: MEMBERS ROLES' and 'Filter table'. A table shows one member: 'dsgcpcert@gmail.com' with the role 'Owner'. There are 'ADD' and 'REMOVE' buttons at the top right of the table.

From there, select the Add option to display another dialog that prompts for usernames and roles (see Figure 3.12).

FIGURE 3.12 Adding a user

The dialog is titled 'Add members to "My First Project"'. It says 'Add members, roles to "My First Project" project' and 'Enter one or more members below. Then select a role for these members to grant them access to your resources. Multiple roles allowed. [Learn more](#)'. It has a 'New members' input field with a question mark icon, a 'Select a role' dropdown, a trash bin icon, and a '+ ADD ANOTHER ROLE' button. At the bottom are 'SAVE' and 'CANCEL' buttons.

Service Accounts

Identities are usually associated with individual users. Sometimes it is helpful to have applications or VMs act on behalf of a user or perform operations that the user does not have permission to perform.

For example, you may have an application that needs to access a database, but you do not want to allow users of the application to access the database directly. Instead, all user requests to the database should go through the application. A service account can be created that has access to the database. That service account can be assigned to the application so the application can execute queries on behalf of users without having to grant database access to those users.

Service accounts are somewhat unusual in that we sometimes treat them as resources and sometime as identities. When we assign a role to a service account, we are treating it as an identity. When we give users permission to access a service account, we are treating it as a resource.

There are two types of service accounts, user-managed service accounts and Google-managed service accounts. Users can create up to 100 service accounts per project. When you create a project that has the Compute Engine API enabled, a Compute Engine service account is created automatically. Similarly, if you have an App Engine application in your project, GCP will automatically create an App Engine service account. Both the Compute Engine and App Engine service accounts are granted editor roles on the projects in which they are created. You can also create custom service accounts in your projects.

Google may also create service accounts that it manages. These accounts are used with various GCP services.

Service accounts can be managed as a group of accounts at the project level or at the individual service account level. For example, if you grant `iam.serviceAccountUser` to a user for a specific project, then that user can manage all service accounts in the project. If you prefer to limit users to manage only specific service accounts, you could grant `iam.serviceAccountUser` for a specific service account.

Service accounts are created automatically when resources are created. For example, a service account will be created for a VM when the VM is created. There may be situations in which you would like to create a service account for one of your applications. In that case, you can navigate to the IAM & admin console and select Service Accounts. From there you can click Create Service Account at the top, as shown in Figure 3.13.

FIGURE 3.13 Service accounts listing in the IAM & admin console

Service accounts		+ CREATE SERVICE ACCOUNT			SHOW INFO PANEL			
Service accounts for project "gcpacel-project"								
A service account represents a Google Cloud service identity, such as code running on Compute Engine VMs, App Engine apps, or systems running outside Google. Learn more								
Filter table								
Email	Name ↑	Description	Key ID	Actions				
agent-logging-service-account@gcpacel-project.iam.gserviceaccount.com	Agent logging service account	Service account for logging	512b64c1afc0a6b4220bebe1a62e5bf7a6729761	⋮				
gcpacel-project@appspot.gserviceaccount.com	App Engine default service account	No keys		⋮				

This brings up a form that prompts for the information needed to create a service account.

Billing

Using resources such as VMs, object storage, and specialized services usually incurs charges. The GCP Billing API provides a way for you to manage how you pay for resources used.

Billing Accounts

Billing accounts store information about how to pay charges for resources used. A billing account is associated with one or more projects. All projects must have a billing account unless they use only free services.

Billing accounts can follow a similar structure to the resource hierarchy. If you are working with a small company, you may have only a single billing account. In that case, all resource costs are charged to that one account. If your company is similar to the example from earlier in the chapter, with finance, marketing, legal, and software development departments, then you may want to have multiple billing accounts. You could have one billing account for each department, but that may not be necessary. If finance, marketing, and legal all pay for their cloud services from the same part of your company's budget, then they could use a single billing account. If software development services are paid from a different part of your company's budget, then it could use a different billing account.

From the main Google Cloud Console, you can navigate to the Billing console (see Figure 3.14), which lists existing billing accounts.

FIGURE 3.14 The main Billing form listing existing billing accounts

The screenshot shows the Google Cloud Platform Billing interface. At the top, there's a blue header bar with the Google Cloud logo and the text "Google Cloud Platform". Below it, a navigation bar has "Billing" selected. To the right of the navigation bar is a dropdown menu labeled "Manage billing accounts". The main content area is titled "Select an organization:" with a dropdown set to "dsgcpcert.com". Below this, there are two tabs: "My billing accounts" (which is selected) and "My projects". Underneath are two buttons: "Create account" and "Show active accounts only". A table follows, with columns: "Billing account name", "Billing account ID", "Status", and "# of projects". One row is visible, showing "My Billing Account" with ID "01FC18-1042CF-AE2C0B", status "Active", and 1 project.

From here, you can create a new billing account, as shown in Figure 3.15.

FIGURE 3.15 The form to create a new billing account

The screenshot shows a "Create a new billing account" form. It has a "Billing" tab at the top left and a "Create a new billing account" title at the top right. The form contains two fields: "Name" with a value of "My Billing Account 1" and "Organization" with a value of "dsgcpcert.com". At the bottom are "Continue" and "Cancel" buttons.

From the Billing overview page, you can view and modify projects linked to billing accounts.

There are two types of billing accounts: self-serve and invoiced. Self-serve accounts are paid by credit card or direct debit from a bank account. The costs are charged automatically. The other type is an invoiced billing account, in which bills or invoices are sent to customers. This type of account is commonly used by enterprises and other large customers.

Several roles are associated with billing. It is important to know them for the exam. The billing roles are as follows:

- Billing Account Creator, which can create new self-service billing accounts
- Billing Account Administrator, which manages billing accounts but cannot create them
- Billing Account User, which enables a user to link projects to billing accounts
- Billing Account Viewer, which enables a user to view billing account cost and transactions

Few users will likely have Billing Account Creator, and those who do will likely have a financial role in the organization. Cloud admins may have Billing Account Administrator to manage the accounts. Any user who can create a project should have Billing Account User so new projects can be linked to the appropriate billing account. Billing Account

Viewer is useful for some, like an auditor who needs to be able to read billing account information but not change it.

Billing Budgets and Alerts

The GCP Billing service includes an option for defining a budget and setting billing alerts. You can navigate to the budget form from the main console menu, selecting Billing and then Budgets & alerts (see Figure 3.16).

FIGURE 3.16 The budget form enables you to have notices sent to you when certain percentages of your budget have been spent in a particular month.

The screenshot shows the 'Create budget' page in the Google Cloud Platform Billing section. The left sidebar lists options: Overview, Budgets & alerts (which is selected and highlighted in blue), Transactions, Billing export, Payment settings, Payment method, and Reports. The main content area has a title 'Set budget'. It explains that budgets can be specified amounts or previous spend, and notes that budget spend resets each month. A note states that setting a budget does not cap resource or API consumption. Below this, there's a 'Budget name' input field, a dropdown for 'Project or billing account' set to 'My Billing Account', and a 'Budget amount' section where 'Specified amount' is chosen. A checkbox for 'Cost after credit' is present. The next section, 'Set budget alerts', allows users to send email notifications for spending thresholds. Three rows are shown: 50%, 90%, and 100%. Each row has a delete 'X' icon and a '+ Add item' button. Below this is a 'Manage notifications' section with a note about Pub/Sub topics and a checkbox for connecting one to the budget. At the bottom are 'Save' and 'Cancel' buttons.

In the budget form, you can name your budget and specify a billing account to monitor. Note that a budget is associated with a billing account, not a project. One or more projects can be linked to a billing account, so the budget and alerts you specify should be based on what you expect to spend for all projects linked to the billing account.

You can specify a particular amount or specify that your budget is the amount spent in the previous month.

With a budget, you can set three alert percentages. By default, three percentages are set: 50 percent, 90 percent, and 100 percent. You can change those to percentages that work best for you. If you'd like more than three alerts, you can click Add Item in the Set Budget Alerts section to add additional alert thresholds.

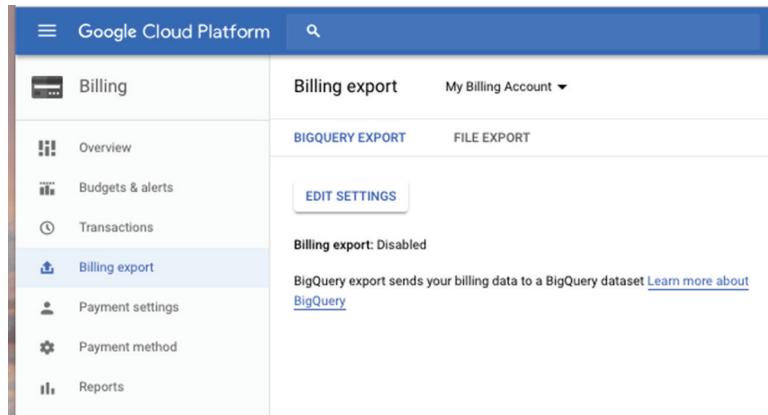
When that percentage of a budget has been spent, it will notify billing administrators and billing account users by email. If you would like to respond to alerts programmatically, you can have notifications sent to a Pub/Sub topic by checking the appropriate box in the Manage Notification sections.

Exporting Billing Data

You can export billing data for later analysis or for compliance reasons. Billing data can be exported to either a BigQuery database or a Cloud Storage file.

To export billing data to BigQuery, navigate to the Billing section of the console and select Billing export from the menu. In the form that appears, select the billing account you would like to export and choose either BigQuery Export or File Export (see Figure 3.17).

FIGURE 3.17 Billing export form



For BigQuery, click Edit Setting. Select the projects you want to include. You will need to create a BigQuery data set to hold the data. Click Go To BigQuery to open a BigQuery form. This will create a Billing export data set, which will be used to hold exported data. (See Figure 3.18.) For additional information on using BigQuery, see Chapter 12.

FIGURE 3.18 Exporting to BigQuery

Billing export My Billing Account ▾

BIGQUERY EXPORT **FILE EXPORT**

BigQuery export sends your billing data to a BigQuery dataset [Learn more about BigQuery](#)

Projects * gcpaces-project

This is where your billing data will be stored. Select a project with BigQuery enabled and with an existing BigQuery dataset.

⚠ You need to create a BigQuery dataset first [GO TO BIGQUERY](#)

Billing export dataset

Billing export will create a table with your billing data in the selected dataset

SAVE **CANCEL**

Alternatively, you can export billing data to a file stored in Cloud Storage. From the Billing Export form, select the File Export tab to display a form as shown in Figure 3.19.

FIGURE 3.19 Exporting billing data to a file

COMPOSE QUERY

Query History
Job History
Scheduled Queries
Transfers

Filter by ID or label ?
AppEngFlex-Project-1

▶ BillingExport

▶ lookerdata

▼ Public Datasets
▶ bigquery-public-data:hacker_news

Dataset Details: BillingExport

Description
Describe this dataset...

Details

Default Table Expiration	Never	Edit
Data Location	US	
Labels	None	Edit

Tables

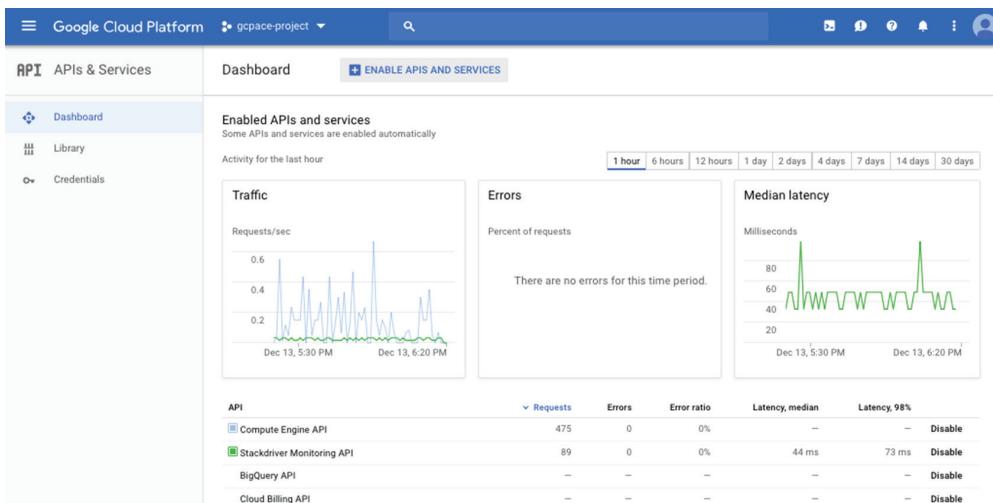
When exporting to a file, you will need to specify a bucket name and a report prefix. You have the option of choosing either the CSV or JSON file format. There may be questions about available file format options, so remember these two options.

Enabling APIs

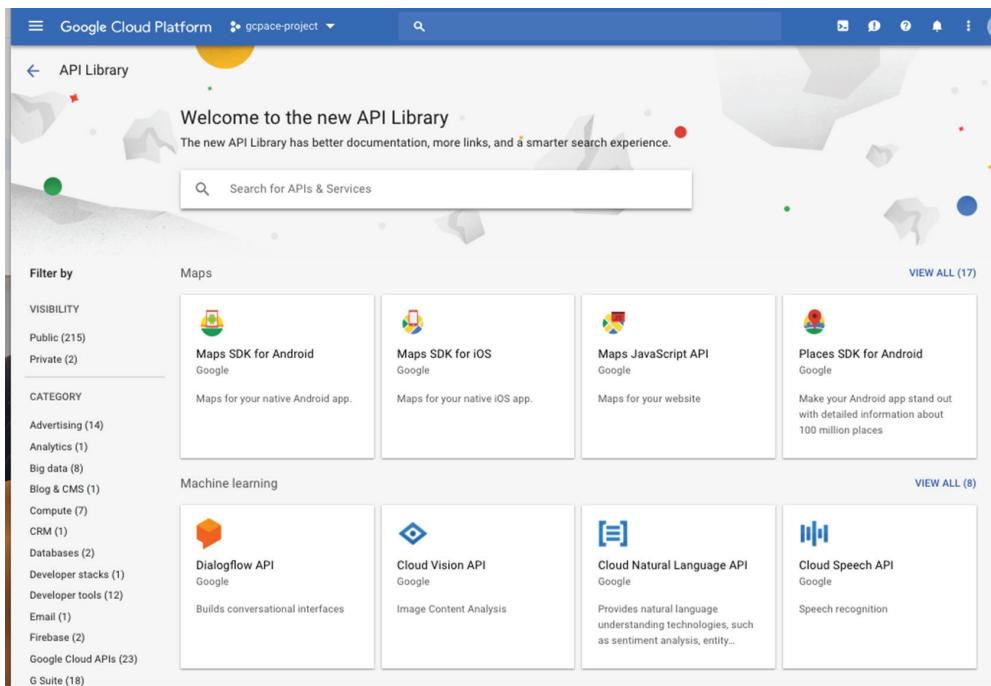
GCP uses APIs to make services programmatically accessible. For example, when you use a form to create a VM or a Cloud Storage bucket, behind the scenes, API functions are executed to create the VM or bucket. All GCP services have APIs associated with them. Most, however, are not enabled by default in a project.

To enable service APIs, you can select APIs & Services from the main console menu. This will display a dashboard, as shown in Figure 3.20.

FIGURE 3.20 An example API services dashboard

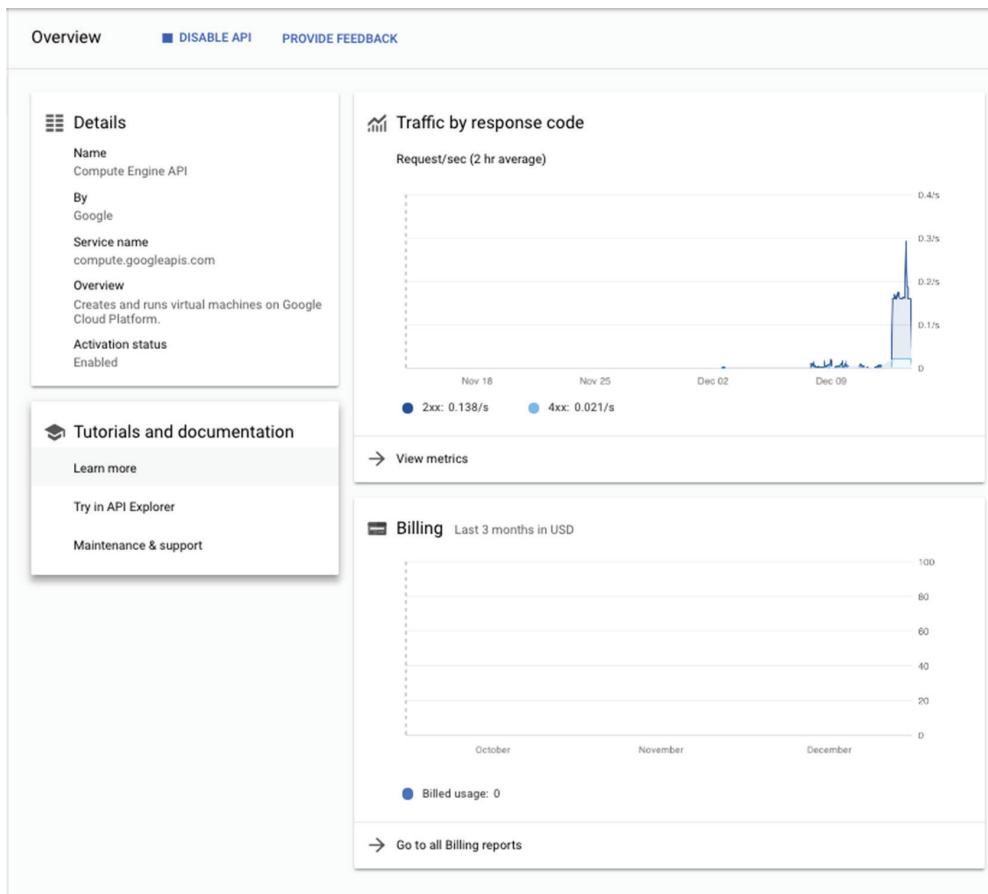


If you click the Enable APIs and Services link, you will see a list of services that you can enable, as shown in Figure 3.21.

FIGURE 3.21 Services that can have their APIs enabled

This form is a convenient way to enable APIs you know you will need. If you attempt an operation that requires an API that is not enabled, you may be prompted to decide if you want to enable the API.

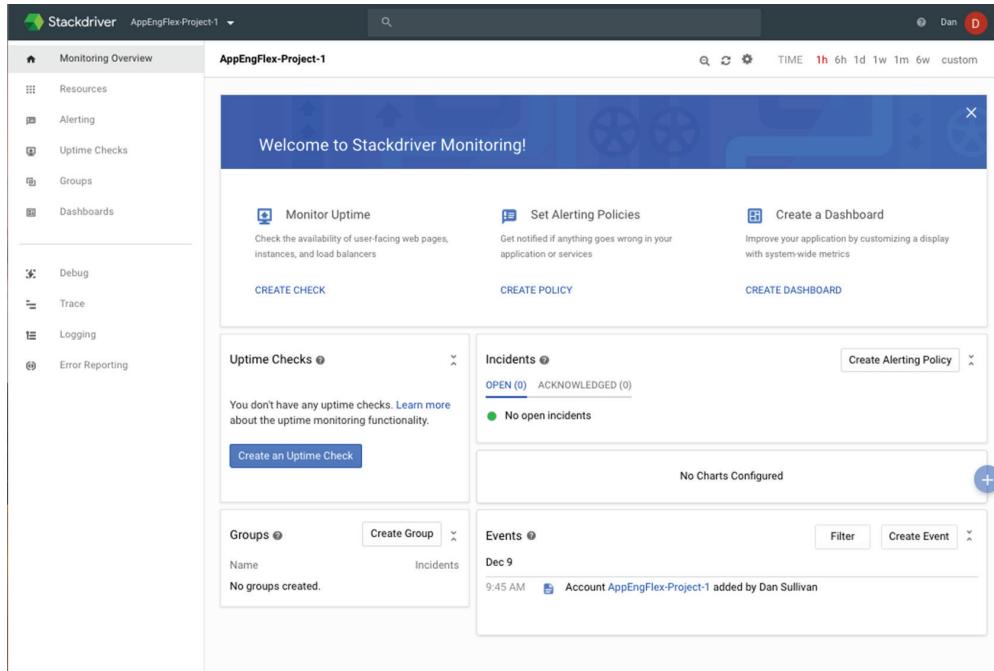
Also, note in Figure 3.20 the list of APIs and their status. Enabled APIs will have a Disable option. You can click that to disable the API. You can also click the name of an API in the list to drill down into details about API usage, as shown in Figure 3.22.

FIGURE 3.22 Details about API usage

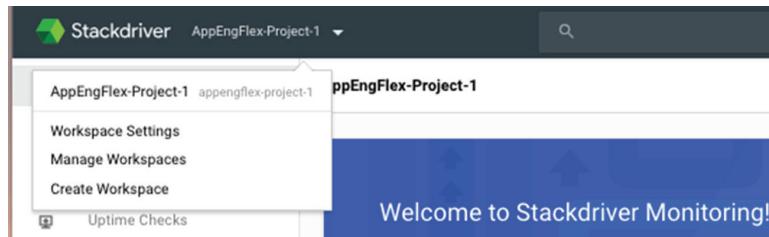
Provisioning Stackdriver Workspaces

When you are setting up organizations and projects, you will spend time on the tasks outlined in this chapter, such as creating identities, assigning roles, and setting up billing accounts. Another thing you should do is create a Stackdriver Workspaces. (These were formerly called Stackdriver accounts, so you may see that term sometimes.)

Stackdriver is a set of services for monitoring, logging, tracing, and debugging applications and resources (see Figure 3.23). For monitoring and logging data to be saved into Stackdriver, you need to create a workspace to save it. You can do this by selecting Stackdriver from the main console menu.

FIGURE 3.23 The main Stackdriver dashboard

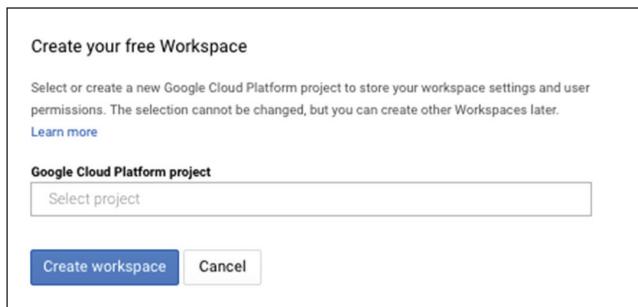
At the top of the dashboard, the name of the current project is displayed. Click the drop-down icon next to the name of the project to display a list of administrative options. One of them is Create Workspace (See Figure 3.24).

FIGURE 3.24 Administrative functions for managing Stackdriver workspaces

If you click Create Workspace, you will see a form like that in Figure 3.25. Select a project from the list that appears when you click in the Google Cloud Platform project box and then click Create Workspace. This will create a workspace and associate it with the project. You will now be able to use monitoring, logging, and other Stackdriver services with your project.

You can find more details on Stackdriver in Chapter 18.

FIGURE 3.25 Create Workspace dialog



Exam Essentials

Understand the GCP resource hierarchy. All resources are organized within your resource hierarchy. You can define the resource hierarchy using one organization and multiple folders and projects. Folders are useful for grouping departments, and other groups manage their projects separately. Projects contain resources such as VMs and cloud storage buckets. Projects must have billing accounts associated with them to use more than free services.

Understand organization policies. Organization policies restrict resources in the resource hierarchy. Policies include constraints, which are rules that define what can or cannot be done with a resource. For example, a constraint can be set to block access to the serial port on all VMs in a project. Also, understand the policy evaluation process and how to override inherited policies.

Understand service accounts and how they are used. Service accounts are identities that are not associated with a specific user but can be assigned to a resource, like a VM. Resources that are assigned a service account can perform operations that the service account has permission to perform. Understand service accounts and how to create them.

Understand GCP Billing. GCP Billing must be enabled to use services and resources beyond free services. Billing associates a billing method, such as a credit card or invoicing information, with a project. All costs associated with resources in a project are billed to the project's billing account. A billing account can be associated with more than one project. You manage your billing through the Billing API.

Know how to enable APIs and create Stackdriver Workspaces. A convenient form lets you enable APIs you know you will need. You can also show a list of APIs and their status. Stackdriver is a set of services for monitoring, logging, tracing, and debugging applications and resources. To monitor and log data to save into Stackdriver, you need to create a workspace.

Review Questions

You can find the answers in the Appendix.

1. You are designing cloud applications for a healthcare provider. The records management application will manage medical information for patients. Access to this data is limited to a small number of employees. The billing department application will have insurance and payment information. Another group of employees will have access billing information. In addition, the billing system will have two components: a private insurance billing system and a government payer billing system. Government regulations require that software used to bill the government must be isolated from other software systems. Which of the following resource hierarchies would meet these requirements and provide the most flexibility to adapt to changing requirements?
 - A. One organization, with folders for records management and billing. The billing folder would have private insurer and government payer folders within it. Common constraints would be specified in organization-level policies. Other policies would be defined at the appropriate folder.
 - B. One folder for records management, one for billing, and no organization. Policies defined at the folder level.
 - C. One organization, with folders for records management, private insurer, and government payer below the organization. All constraints would be specified in organization-level policies. All folders would have the same policy constraints.
 - D. None of the above.
2. When you create a hierarchy, you can have more than one of which structure?
 - A. Organization only
 - B. Folder only
 - C. Folder and project
 - D. Project only
3. You are designing an application that uses a series of services to transform data from its original form into a format suitable for use in a data warehouse. Your transformation application will write to the message queue as it processes each input file. You don't want to give users permission to write to the message queue. You could allow the application to write to the message queue by using which of the following?
 - A. Billing account
 - B. Service account
 - C. Messaging account
 - D. Folder

4. Your company has a number of policies that need to be enforced for all projects. You decide to apply policies to the resource hierarchy. Not long after you apply the policies, an engineer finds that an application that had worked prior to implementing policies is no longer working. The engineer would like you to create an exception for the application. How can you override a policy inherited from another entity in the resource hierarchy?
 - A. Inherited policies can be overridden by defining a policy at a folder or project level.
 - B. Inherited policies cannot be overridden.
 - C. Policies can be overridden by linking them to service accounts.
 - D. Policies can be overridden by linking them to billing accounts.
5. Constraints are used in resource hierarchy policies. Which of the following are types of constraints allowed?
 - A. Allow a specific set of values
 - B. Deny a specific set of values
 - C. Deny a value and all its child values
 - D. Allow all allowed values
 - E. All of the above
6. A team with four members needs you to set up a project that needs only general permissions for all resources. You are granting each person a primitive role for different levels of access, depending on their responsibilities in the project. Which of the following are not included as primitive roles in Google Cloud Platform?
 - A. Owner
 - B. Publisher
 - C. Editor
 - D. Viewer
7. You are deploying a new custom application and want to delegate some administration tasks to DevOps engineers. They do not need all the privileges of a full application administrator, but they do need a subset of those privileges. What kind of role should you use to grant those privileges?
 - A. Primitive
 - B. Predefined
 - C. Advanced
 - D. Custom
8. An app for a finance company needs access to a database and a Cloud Storage bucket. There is no predefined role that grants all the needed permissions without granting some permissions that are not needed. You decide to create a custom role. When defining custom roles, you should follow which of the following principles?
 - A. Rotation of duties
 - B. Least principle

- C. Defense in depth
 - D. Least privilege
9. How many organizations can you create in a resource hierarchy?
- A. 1
 - B. 2
 - C. 3
 - D. Unlimited
10. You are contacted by the finance department of your company for advice on how to automate payments for GCP services. What kind of account would you recommend setting up?
- A. Service account
 - B. Billing account
 - C. Resource account
 - D. Credit account
11. You are experimenting with GCP for your company. You do not have permission to incur costs. How can you experiment with GCP without incurring charges?
- A. You can't; all services incur charges.
 - B. You can use a personal credit card to pay for charges.
 - C. You can use only free services in GCP.
 - D. You can use only serverless products, which are free to use.
12. Your DevOps team has decided to use Stackdriver monitoring and logging. You have been asked to set up Stackdriver workspaces. When you set up a Stackdriver workspace, what kind of resource is it associated with?
- A. A Compute Engine instance only
 - B. A Compute Engine instance or Kubernetes Engine cluster only
 - C. A Compute Engine instance, Kubernetes Engine cluster, or App Engine app
 - D. A project
13. A large enterprise is planning to use GCP across a number of subdivisions. Each subdivision is managed independently and has its own budget. Most subdivisions plan to spend tens of thousands of dollars per month. How would you recommend they set up their billing account(s)?
- A. Use a single self-service billing account.
 - B. Use multiple self-service billing accounts.
 - C. Use a single invoiced billing account.
 - D. Use multiple invoiced billing accounts.

- 14.** An application administrator is responsible for managing all resources in a project. She wants to delegate responsibility for several service accounts to another administrator. If additional service accounts are created, the other administrator should manage those as well. What is the best way to delegate privileges needed to manage the service accounts?
- A.** Grant `iam.serviceAccountUser` to the administrator at the project level.
 - B.** Grant `iam.serviceAccountUser` to the administrator at the service account level.
 - C.** Grant `iam.serviceProjectAccountUser` to the administrator at the project level.
 - D.** Grant `iam.serviceProjectAccountUser` to the administrator at the service account level.
- 15.** You work for a retailer with a large number of brick and mortar stores. Every night the stores upload daily sales data. You have been tasked with creating a service that verifies the uploads every night. You decide to use a service account. Your manager questions the security of your proposed solution, particularly about authenticating the service account. You explain the authentication mechanism used by service accounts. What authentication mechanism is used?
- A.** Username and password
 - B.** Two-factor authentication
 - C.** Encrypted keys
 - D.** Biometrics
- 16.** What objects in GCP are sometimes treated as resources and sometimes as identities?
- A.** Billing accounts
 - B.** Service accounts
 - C.** Projects
 - D.** Roles
- 17.** You plan to develop a web application using products from the GCP that already include established roles for managing permissions such as read-only access or the ability to delete old versions. Which of the following roles offers these capabilities?
- A.** Primitive roles
 - B.** Predefined roles
 - C.** Custom roles
 - D.** Application roles
- 18.** You are reviewing a new GCP account created for use by the finance department. An auditor has questions about who can create projects by default. You explain who has privileges to create projects by default. Who is included?
- A.** Only project administrators
 - B.** All users
 - C.** Only users without the role `resourcemanager.projects.create`
 - D.** Only billing account users

- 19.** How many projects can be created in an account?
- A.** 10
 - B.** 25
 - C.** There is no limit.
 - D.** Each account has a limit determined by Google.
- 20.** You are planning how to grant privileges to users of your company's GCP account. You need to document what each user will be able to do. Auditors are most concerned about a role called Organization IAM roles. You explain that users with that role can perform a number of tasks, which include all of the following except which one?
- A.** Defining the structure of the resource hierarchy
 - B.** Determining what privileges a user should be assigned
 - C.** Defining IAM policies over the resource hierarchy
 - D.** Delegating other management roles to other users

Chapter 4



Introduction to Computing in Google Cloud

THIS CHAPTER COVERS THE FOLLOWING OBJECTIVES OF THE GOOGLE ASSOCIATE CLOUD ENGINEER CERTIFICATION EXAM:

- ✓ 2.2 Planning and configuring compute resources



In this chapter, you will learn about each of the compute options available in Google Cloud Platform (GCP) and when to use them. We will also discuss preemptible virtual machines and when they can help reduce your overall computing costs.

Compute Engine

Compute Engine is a service that provides VMs that run on GCP. We usually refer to a running VM as an *instance*. When you use Compute Engine, you create and manage one or more instances.

Virtual Machine Images

Instances run images, which contain operating systems, libraries, and other code. You may choose to run a public image provided by Google (Figure 4.1). Both Linux and Windows images are available. In addition to the images maintained by Google, there are other public images provided by open source projects or third-party vendors.

FIGURE 4.1 A subset of operating system images available in Compute Engine

Boot disk

Select an image or snapshot to create a boot disk; or attach an existing disk

[OS images](#) [Application images](#) [Custom images](#) [Snapshots](#) [Existing disks](#)

- SUSE Linux Enterprise Server 12 SP2 For SAP x86_64
built on 2018-08-16
- SUSE Linux Enterprise Server 12 SP3 For SAP x86_64
built on 2018-08-14
- SUSE Linux Enterprise Server 12 SP4 For SAP x86_64
built on 2018-12-12
- SUSE Linux Enterprise Server 15 For SAP x86_64
built on 2018-08-16
- Windows Server version 1709 Datacenter Core for Containers
Server Core, x64 built on 20181113
- Windows Server version 1709 Datacenter Core
Server Core, x64 built on 20181113
- Windows Server version 1803 Datacenter Core for Containers
Server Core, x64 built on 20181113
- Windows Server version 1803 Datacenter Core
Server Core, x64 built on 20181113
- Windows Server 2008 R2 Datacenter
Server with Desktop Experience, x64 built on 20181113
- Windows Server 2012 R2 Datacenter Core
Server Core, x64 built on 20181113
- Windows Server 2012 R2 Datacenter
Server with Desktop Experience, x64 built on 20181113
- Windows Server 2016 Datacenter Core
Server Core, x64 built on 20181113
- Windows Server 2016 Datacenter
Server Core, x64 built on 20181113

The public images include a range of operating systems, such as CentOS, Container Optimized OS from Google, Debian, Red Hat Enterprise Linux, SUSE Enterprise Linux Server, Ubuntu, and Windows Server.

If there is no public image that meets your needs, you can create a custom image from a boot disk or by starting with another image. To create a VM from the console, navigate to Compute Engine and then to VM Instances. You will see a screen similar to Figure 4.2.

FIGURE 4.2 Creating a VM in Compute Engine

The screenshot shows the Compute Engine interface with the 'VM instances' tab selected. On the left, a sidebar lists 'VM instances', 'Instance groups', 'Instance templates', 'Sole tenant nodes', and 'Disks'. The main area displays a table of VM instances with one entry:

Name	Zone	Recommendation	Internal IP	External IP	Connect
myvm	us-west1-b		10.138.0.2 (nic0)	35.247.42.4	SSH

From there, click Create Instance to create a VM. Choose an image that is close to what you need and create the VM. Then make any changes you need to the image, such as installing libraries or other software packages. Once you have created the VM and made any changes you'd like, navigate to the Compute Engine menu in Google Cloud Console and select Snapshots, as shown in Figure 4.3.

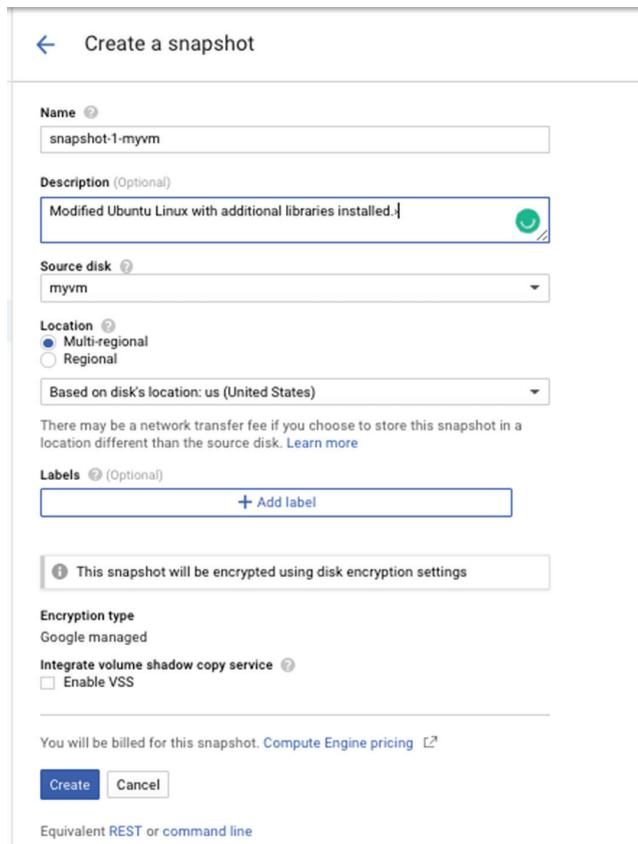
FIGURE 4.3 The first step in creating a snapshot

The screenshot shows the Compute Engine interface with the 'Solutions' tab selected. On the left, a sidebar lists 'VM instances', 'Instance groups', 'Instance templates', 'Sole tenant nodes', 'Disks', 'Solutions' (which is selected), 'Images', 'TPUs', 'Committed use discounts', and 'Metadata'. The main area contains a box titled 'Compute Engine Snapshots' with the following text: 'You can take a snapshot of a Compute Engine persistent disk to quickly back up the disk so you can recover lost data, transfer contents to a new disk, or make static data available to multiple nodes. Learn more'. Below the text is a blue button labeled 'Create snapshot'.

Click Create Snapshot. In the form that appears, you can specify a name for your snapshot, a description, and, most importantly, the disk that is the source for the snapshot. In Figure 4.4, the boot disk from a VM called `myvm` is selected. After selecting

options, click the Create button to save the snapshot, which can then be used as the image for other VMs.

FIGURE 4.4 Creating a snapshot in Compute Engine



Custom images are especially useful if you have to configure an operating system and install additional software on each instance of a VM that you run. Instead of repeatedly configuring and installing software for each instance, you could configure and install once and then create a custom image from the boot disk of the instance. You then specify that custom image when you start other instances, which will have the configuration and software available without any additional steps.

There may be cases where you have a custom image in your local environment or data center. You can import such an image using the virtual disk import tool provided by Google. This utility is part of the `gcloud` command-line tool, which will be described in more detail in the next chapter.

Custom images must be compatible with GCP. At the time of writing, the following base operating systems are available to build custom images that will run in Compute Engine:

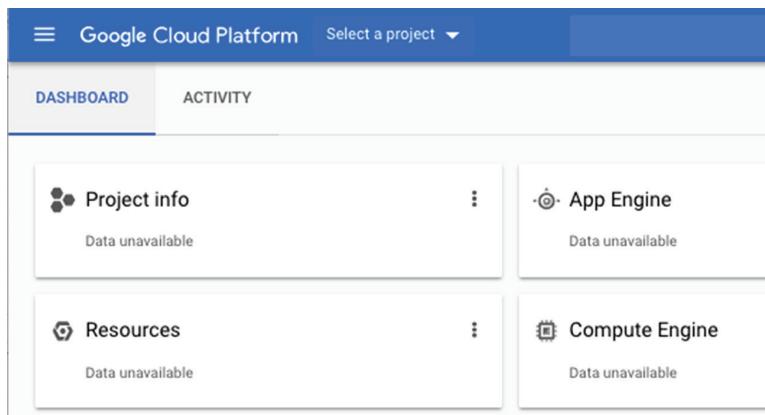
- Linux operating systems
 - CentOS 6
 - CentOS 7
 - Debian 8
 - Debian 9
 - Red Hat Enterprise Linux 6
 - Red Hat Enterprise Linux 7
 - Ubuntu 14.04 LTS
 - Ubuntu 15.04 LTS
- Windows operating systems
 - Windows Server 2008 R2
 - Windows Server 2012 R2
 - Windows Server 2012 R2 Core
 - Windows Server 2016
 - Windows Server 2016 Core

Virtual Machines Are Contained in Projects

When you create an instance, you specify a project to contain the instance. As you may recall, projects are part of the GCP resource hierarchy. Projects are the lowest-level structure in the hierarchy. Projects allow you to manage related resources with common policies.

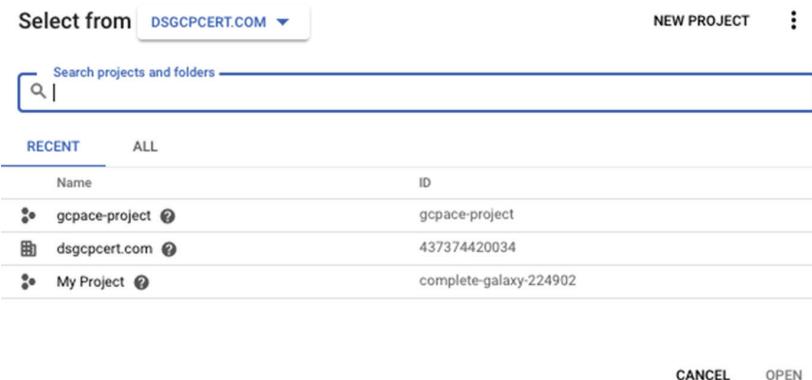
When you open Google Cloud Console, you will notice at the top of the form either the name of a project or the phrase Select A Project, as shown in Figure 4.5.

FIGURE 4.5 The current project name or the option to select one is displayed in Google Cloud Console.



When you choose Select a Project, a form like that in Figure 4.6 appears. From there, you can select the project you want to store your resources, including VMs.

FIGURE 4.6 Choosing a project from existing projects in an account



Virtual Machines Run in a Zone and Region

In addition to having a project, VM instances have a zone assigned. Zones are data center-like resources, but they may be comprised of one or more closely coupled data centers. They are located within regions. A *region* is a geographical location, such as `asia-east1`, `europe-west2`, and `us-east4`. The zones within a region are linked by low-latency, high-bandwidth network connections.

You specify a region and a zone when you create a VM. As you can see in Figure 4.7, the Create VM form includes drop-down lists from which you can select the region and zone.

FIGURE 4.7 Selecting a region and zone in the Create VM form

The screenshot shows the 'Create VM' form. It has several input fields and dropdown menus:

- Name**: A text input field containing the value 'instance-1'.
- Region**: A dropdown menu set to 'us-west2 (Los Angeles)'.
- Zone**: A dropdown menu set to 'us-west2-a'.
- Machine type**: A section with a note 'Customize to select cores, memory and GPUs.' It includes a dropdown for '1 vCPU', a text input for '3.75 GB memory', and a 'Customize' button.
- Upgrade your account**: A note at the bottom of the machine type section.

You may want to consider several factors when choosing where to run your VM, including the following:

- Cost, which can vary between regions.
- Data locality regulations, such as keeping data about European Union citizens in the European Union.
- High availability. If you are running multiple instances, you may want them in different zones and possibly different regions. If one of the zones or regions become inaccessible, the instances in other zones and regions can still provide services.
- Latency, which is important if you have users in different parts of the world. Keeping instances and data geographically close to application users can help reduce latency.
- Need for specific hardware platforms, which can vary by region. For example, at the time of writing europe-west1 has Intel Xeon E5, also known as Sandy Bridge, platforms, but Europe-west2 does not.

Users Need Privileges to Create Virtual Machines

To create Compute Engine resources in a project, users must be team members on the project or a specific resource and have appropriate permissions to perform specific tasks. Users can be associated with projects as follows:

- Individual users
- A Google group
- A G Suite domain
- A service account

Once a user or set of users is added to a project, you can assign permissions by granting roles to the user or set of users. This process is explained in detail in Chapter 17. Predefined roles are especially useful because they group permissions that are often needed together for a user to carry out a set of tasks. Here are some examples of predefined roles:

Compute Engine Admin Users with this role have full control over Compute Engine instances.

Compute Engine Network Admin Users with this role can create, modify, and delete most networking resources, and it provides read-only access to firewall rules and SSL certifications. This role does not give the user permission to create or alter instances.

Compute Engine Security Admin Users with this role can create, modify, and delete SSL certificates and firewall rules.

Compute Engine Viewer Users with this role can get and list Compute Engine resources but cannot read data from those resources.

When privileges are granted to users at the project level, then those permissions apply to all resources within a project. For example, if a user is granted the Compute Engine Admin role at the project level, then that person can administer all Compute Engine instances in the project. Figure 4.8 shows an example listing of users and roles.

FIGURE 4.8 An example listing of users and roles

The screenshot shows the Google Cloud IAM interface. At the top, there are buttons for 'IAM', '+ ADD' (with a person icon), and '- REMOVE' (with a person icon). Below this, the title 'Permissions for organization "dsgcpcert.com"' is displayed, followed by a note: 'These permissions affect this organization and all of its resources. [Learn more](#)'. Under 'View By:', the 'MEMBERS' tab is selected, showing a list of members and their roles. A 'Filter table' button is available. The table has columns: Type, Member ↑, Name, Role, and Inheritance. There are two rows for the member 'dan@dsgcpcert.com':

Type	Member ↑	Name	Role	Inheritance
<input type="checkbox"/>		dan@dsgcpcert.com	Dan Sullivan Organization Role Administrator Organization Policy Administrator	
<input type="checkbox"/>		dsgcpcert.com	Billing Account Creator Project Creator	

An alternative way to grant permissions is to attach IAM policies directly to resources. In this way, privileges can be tailored to specific resources instead of for all resources in a project. For example, you could specify that user Alice has the Compute Engine Admin role on one instance and Bob has the same role on another instance. Alice and Bob would be able to administer their own VM instances, but they could not administer other instances.

Preemptible Virtual Machines

Consider if you have a workload that is the opposite of needing high availability. Preemptible VMs are short-lived compute instances suitable for running certain types of workloads—particularly for applications that perform financial modeling, rendering, big data, continuous integration, and web crawling operations. These VMs offer the same configuration options as regular compute instances and persist for up to 24 hours. If an application is fault-tolerant and can withstand possible instance interruptions (with a 30 second warning), then using preemptible VM instances can reduce Google Compute Engine costs significantly.



Real World Scenario

Reformatting Images

A mobile application that allows users to upload photos also reformats those images into a variety of formats. While it is important for the original image to upload quickly, there is no pressing need to have the other formats created quickly. If it were to take several

minutes, that would still meet the application requirements. This is a good use case for preemptible machines.

When a file is uploaded, it triggers a cloud function (described in detail in Chapter 10), which starts the reformatting process on a preemptible VM. When the reformatting is complete, the reformatted image is written to storage. If the machine shuts down while reformatting an image, that image could be reformatted again when another VM starts up. When the reformatting application starts, it checks for any images that do not have all reformatted versions. If some images are missing some reformatted options, it can start reformatting those. This process can run at regular intervals to check whether any images have not been reformatted. In this way, we can use preemptible VMs and still meet the service-level objectives.

Some big data analysis jobs run on clusters of servers running software like Hadoop and Spark. The platforms are designed to be resilient to failure. If a node goes down in the middle of a job, the platform detects the failure and moves workload to other nodes in the server. You may have analytic jobs that are well served by a combination of reliable VMs and preemptible VMs. With some percentage of reliable VMs, you know you can get your jobs processed within your time constraints, but if you add low-cost, preemptible VMs, you can often finish your jobs faster and at lower overall cost.

Limitations of Preemptible Virtual Machines

As you decide where to use preemptible VMs, keep in mind their limitations and differences compared to conventional VM instances in GCP. Preemptible VMs have the following characteristics:

- May terminate at any time. If they terminate within 10 minutes of starting, you will not be charged for that time.
- Will be terminated within 24 hours.
- May not always be available. Availability may vary across zones and regions.
- Cannot migrate to a regular VM.
- Cannot be set to automatically restart.
- Are not covered by any service level agreement (SLA).

Custom Machine Types

Compute Engine has more than 25 predefined machine types grouped into standard types, high-memory machines, high-CPU machines, shared core type, and memory-optimized machines. These predefined machine types vary in the number of virtual CPUs (vCPUs) and amount of memory. Here are some examples:

- n1-standard-1 has 1 vCPU and 3.75GB of memory.
- n1-standard-32 has 32 vCPUs and 120GB of memory.
- n1-higmem-32 has 32 vCPUs and 208GB of memory.
- n1-highcpu-32 has 32 vCPUs and 28.8GB of memory.

The predefined options for VMs will meet the needs of many use cases, but there may be times where your workload could run more cost effectively and faster on a configuration that is not already defined. In that case, you may want to use a custom machine type.

To create a custom image, select the Create VM option in the console. Click the Customize link in the Machine Type section. This expands the Machine Type section, as shown in Figure 4.9. From there you can adjust the sliders to increase or decrease the number of CPUs and the amount of memory you require.

FIGURE 4.9 Customizing a VM by adjusting the number of CPUs and the amount of memory



Custom machine types can have between 1 and 64 vCPUs and up to 6.5GB of memory per vCPU. The price of a custom configuration is based on the number of vCPUs and the memory allocated.

Use Cases for Compute Engine Virtual Machines

Compute Engine is a good option when you need maximum control over VM instances. With Compute Engine, you can do the following:

- Choose the specific image to run on the instance.
- Install software packages or custom libraries.
- Have fine-grained control over which users have permissions on the instance.
- Have control over SSL certificates and firewall rules for the instance.

Relative to other computing services in GCP, Google Compute Engine provides the least amount of management. Google does provide public images and a set of VM configurations, but you as an administrator must make choices about which image to use, the number of CPUs, the amount of memory to allocate, how to configure persistent storage, and how to configure network configurations.

In general, the more control over a resource you have in GCP, the more responsibility you have to configure and manage the resource.

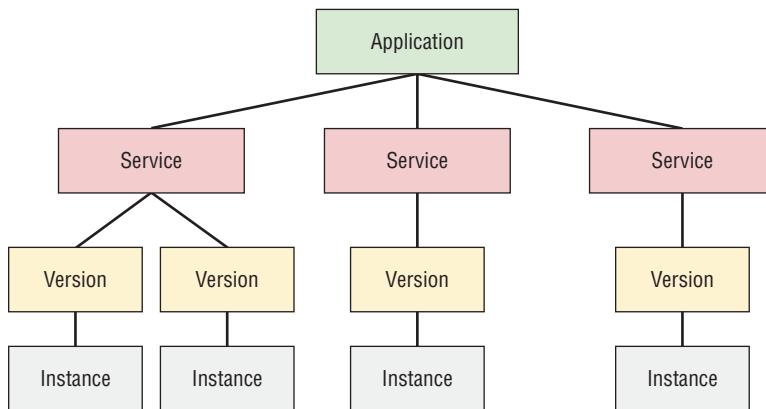
App Engine

App Engine is a PaaS compute service that provides a managed platform for running applications. When you use App Engine, your focus is on your application and not on the VMs that run the application. Instead of configuring VMs, you specify some basic resource requirements along with your application code, and Google will manage the resources needed to run the code. This means that App Engine users have less to manage, but they also have less control over the compute resources that are used to execute the application.

Like VM instances, applications in App Engine are created within a project.

Structure of an App Engine Application

App Engine applications have a common structure, and they consist of services. Services provide a specific function, like computing sales tax in a retail web application or updating inventory as products are sold on a site. Services have versions, and this allows multiple versions to run at one time. Each version of a service runs on an instance that is managed by App Engine (see Figure 4.10).

FIGURE 4.10 The structure of an App Engine application

The number of instances used to provide an application depends on your configuration for the application and the current load on the application. As the load increases, Google can add more instances to meet the need. Similarly, if the load lessens, instances can be shut down to save on the cost of unutilized instances. This kind of autoscaling is available with dynamic instances.

In addition to dynamic instances, App Engine also provides resident instances. These instances run continually. You can add or remove resident instances manually.

When the number of deployed instances changes frequently, it can be difficult to estimate the costs of running instances. Fortunately, GCP allows users to set up daily spending limits as well as create budgets and set alarms.

App Engine Standard and Flexible Environments

App Engine provides two types of runtime environments: standard and flexible. The standard environment provides language runtimes, while the flexible environment is a more generalized container execution platform.

App Engine Standard Environment

The standard environment is the original App Engine environment. It consists of a preconfigured, language-specific runtime. There are currently two generations of the standard environment. The second generation improves on the performance of the first generation and has fewer limitations.

Currently, App Engine standard environment users can choose from the following supported languages: