Become GCP Cloud Digital Leader Certified in 7 Days

Getting to Know Google Cloud

Abhinav Krishna Kaiser

Apress®

Become GCP Cloud Digital Leader Certified in 7 Days

Getting to Know Google Cloud

Abhinav Krishna Kaiser

Apress®

Become GCP Cloud Digital Leader Certified in 7 Days: Getting to Know Google Cloud

Abhinav Krishna Kaiser
Bengaluru, Karnataka, India

ISBN-13 (pbk): 979-8-8688-0437-3 ISBN-13 (electronic): 979-8-8688-0438-0
https://doi.org/10.1007/979-8-8688-0438-0

Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Celestin Suresh John
Development Editor: James Markham
Coordinating Editor: Gryffin Winkler
Copy Editor: Kezia Endsley

Cover designed by eStudioCalamar

Cover image by Gianni Crestani from Pixabay

Distributed to the book trade worldwide by Apress Media, LLC, 1 New York Plaza, New York, NY 10004, U.S.A. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail booktranslations@springernature.com; for reprint, paperback, or audio rights, please e-mail bookpermissions@springernature.com.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at http://www.apress.com/bulk-sales.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub (https://github.com/Apress). For more detailed information, please visit https://www.apress.com/gp/services/source-code.

If disposing of this product, please recycle the paper

Table of Contents

About the Author

Abhinav Krishna Kaiser is a highly accomplished professional working as a partner at a prestigious consulting firm, where he plays a pivotal role in leading digital transformation programs for clients across diverse sectors. He is part of the Distinguished Member of Technical Staff (DMTS) cadre, which represents a select group of best-in-class technologists. With a proven track record in the industry, Abhinav is recognized for his expertise in guiding organizations through complex and innovative changes to stay ahead of the curve in today's dynamic business environment.

Abhinav spearheads various digital transformation initiatives, demonstrating a keen understanding of the unique challenges and opportunities presented by different industries. His leadership is characterized by a strategic and holistic approach, ensuring that clients not only adapt to current market trends but also position themselves for future success. His portfolio includes successfully steering multiple digital transformation programs, showcasing his ability to navigate and drive change in organizations of varying sizes and complexities. His hands-on experience in implementing cutting-edge technologies and methodologies has contributed to the enhanced efficiency and competitiveness of his clients.

In addition, Abhinav is a multifaceted professional with a prolific career as an accomplished writer. He boasts an impressive literary portfolio of five published books, each delving into the intricacies of digital transformation, DevOps, and ITIL. Abhinav's written works serve as authoritative guides, offering valuable insights and practical solutions to professionals navigating the complexities of modern business and technology landscapes.

Beyond his contributions in the written domain, Abhinav is a panel speaker, captivating audiences with his expertise at industry conferences and events. His commitment to knowledge-sharing extends to digital platforms, where he actively engages as a YouTuber and blogger. Through these media, he imparts knowledge, shares best practices, and explores emerging trends, reaching a wider audience eager to enhance their understanding of digital transformation, DevOps, and ITIL. As a thought leader in the digital space, Abhinav's online presence further solidifies his influence, making him a go-to resource for professionals and enthusiasts alike.

About the Technical Reviewer

Ravi Mishra is a distinguished multi-cloud architect with over 15 years of rich experience in the IT industry. His journey commenced as a network engineer, where he laid the foundation for his illustrious career. Driven by an insatiable curiosity and a relentless pursuit of excellence, Ravi transitioned into the dynamic realm of cloud computing. Over time, he honed his expertise across various cloud platforms, including AWS, Azure, GCP, Ali, and Oracle, establishing himself as a seasoned authority in these fields.

A strong academic foundation complements Ravi's practical experience. He holds an electronics engineering degree and a postgraduate diploma in IT project management, providing him with a solid understanding of the technical and managerial aspects of IT projects. Throughout his career, Ravi has collaborated with various global MNCs, demonstrating his ability to adapt to diverse work environments.

Ravi's relentless quest for knowledge is evidenced by his remarkable achievement of acquiring over 43 cloud certifications, a testament to his dedication and proficiency in the domain. As a seasoned Microsoft Certified Trainer for the past eight years, he has imparted invaluable knowledge to over 1,00,000 aspiring professionals, leaving an indelible mark on the industry.

An enthusiastic advocate of emerging technologies, Ravi is a sought-after speaker at various public forums, where he shares his insights on cutting-edge technologies like Copilot, AI, DevOps, Terraform, and cloud security. Additionally, he is the esteemed author of *HashiCorp Infrastructure Automation Certification Guide*, a comprehensive resource that delves into infrastructure automation across AWS, Azure, and GCP using Terraform.

In his free time, Ravi pursues his passions for cooking, badminton, and running marathons, demonstrating a healthy balance between professional excellence and personal well-being. Connect with Ravi on LinkedIn (`inmishrar`) to learn more about his journey and insights on the ever-evolving world of technology.

Introduction

The title of this book could be misleading. Yes, it is meant to help enthusiasts earn Google Cloud Platform's Cloud Digital Leader certification. However, this book is not restricted to digital leaders. It is also a 101 course on anything to do with the field of IT, the cloud, and DevOps, among other foundational concepts that add to the foundation of digital transformation. Whether you're a seasoned IT professional, an aspiring digital leader, or an executive driving organizational change, this book is your definitive roadmap to success in the cloud. I could have very well called the book *Getting Started with IT along with the Google Cloud Platform,* because I have broken down the foundational elements of IT (databases, serverless functions, artificial intelligence, and so on) into easily consumable chunks. It's like baby food!

Dialing back, the premise of this book started with my obsession of digital transformation and Google's interface-friendly ecosystem for the cloud. I must confess that I picked up Google Cloud Platform (GCP) after I had played my hand in Amazon Web Services (AWS) and MS Azure. No disrespect to the other two major cloud service providers, but the simplicity behind GCP was awe-inspiring. It demanded a book, especially when I was in the midst of busiest time of my work life.

The beauty of digital transformation is that it has no boundaries. Its currencies are innovation and agility, and the organizations that showcase these qualities are best positioned to stay ahead of the curve and enjoy unprecedented growth. This is where the cloud comes in, by helping them with the latest innovative products that can be bent at will to meet the organization's objectives. In my previous book, *Reinventing ITIL and DevOps with Digital Transformation*, I introduced the *Battle Tank* framework. It is not prescriptive, but rather a guide for organizations to plan and embark on the digital transformation journey. If you are a digital leader or an enthusiast looking to explore the digital transformation world further, I highly recommend that book, which builds on the concepts of ITIL and DevOps to unravel digital transformation.

In the pages that follow, I dive into the core components of Google Cloud Platform, exploring everything from foundational concepts to semi-advanced use cases. You'll learn how to leverage GCP's vast array of products to drive efficiency, enhance collaboration, and unlock new opportunities for growth.

The book is structured into 13 logical chapters, and each chapter delves into an aspect of the cloud. The first four chapters focus on building the narrative on the cloud, the GCP basics, and digital transformation. Beginning with Chapter 5, GCP capabilities are bundled into chapters for ease of reading and to introduce relevance. For example, Chapter 5 covers the concept of virtual machines, including the various products that exist in GCP that support virtual machines. Chapter 6 covers containers, followed by serverless computing in Chapter 7. These three chapters introduce GCP products on infrastructure.

Chapter 8 shifts the focus to application development with the modernization products. The biggest chapter in the book is Chapter 9 and it covers data. This is followed by the exciting world of machine learning and AI, covered in Chapter 10. Financial governance and security on GCP are covered in Chapters 11 and 12, respectively. Finally, the book concludes with the topics of monitoring, observability, DevOps, and Site Reliability Engineering (SRE).

Google has partnered with Kryterion to enable users to register for this exam. You can take the exam at a physical center or remotely, which is proctored. At the time of this writing, the Google Cloud Platform: Cloud Digital Leader certification exam costs $99 USD. There are between 60-70 questions on the exam and you have 90 minutes to answer them. You are expected to choose the best answer from the options. I don’t believe that you need all 90 minutes to answer the questions, but use your time wisely to revise your answers until the last minute. To pass the exam, you need to get 70 percent or more correct. Although the exam is fairly straightforward, the manner in which the questions are posed can be confusing.

I wish you happy learning as you read this book, and I hope that you reach great heights in your career. All the very best!

DAY 1

Approximate Study Time: 1 hour and 26 minutes

Chapter 1 - 48 minutes

Chapter 2 - 38 minutes

CHAPTER 1

Cloud Computing Overview

If I had gone into hibernation 15 years ago and woke up today, I wouldn't recognize many aspects of the world. For instance, when flying on an airplane, we no longer need to carry printed boarding passes or flight tickets. So much has gone digital. To wish a friend on their birthday, we can simply order a cake to be delivered to their house in a couple of minutes. In India as in elsewhere, everything is now entirely digital. Even many street vendors use QR codes to accept payment for purchases.

The changes, all good from the perspective of dynamism, are possible because data is available everywhere on any device—be it a smartphone or a laptop. The ecosystem of dynamism has been through its own version of evolution, which is discussed later in this chapter.

The genesis was the cloud. The cloud made it possible for new technologies to take shape. Varied use cases, ideas, and solutions shaped new user behaviors. So what do I mean by the cloud?

In principle, cloud computing refers to unlimited (or an abundance of) computing resources available on demand. Computing resources include servers, routers, storage, and containers, among others. Depending on the requirements, these computer resources can be scaled in a span of minutes if not seconds. Many people no longer worry about infrastructure constraints. Say, for example, I needed to increase the RAM on a server hosted in a datacenter. Prior to the cloud, I had to plan for a downtime,

A. K. Kaiser, *Become GCP Cloud Digital Leader Certified in 7 Days*,
https://doi.org/10.1007/979-8-8688-0438-0_1

arrange for an engineer to bring the compatible RAM, coordinate with the facilities, and then manage the changes along with the post-change activities. With the cloud, it's as simple as it gets. Through a console, I can increase the RAM, and it is added without any of the pains of going through a physical set of change activities. I have the option of scaling up vertically or horizontally with the click of a few buttons, and I get the option of a load balancer and active backups. With cloud computing, managing and planning infrastructure has become seamless. Cloud computing has helped companies focus on the strategic aspects of their business rather than being bothered with operations, which has the potential to bring down the business.

Cloudification is rampant in organizations, and it continues to grow. Self-managed datacenters are becoming old fashioned and the benefits of the cloud outweigh and out-compare the rationale of owning and managing private datacenters.

Figure 1-1 shows the increase in cloud spending from 2021 to 2023 pointing towards the migration from on-premises to cloud infrastructure.

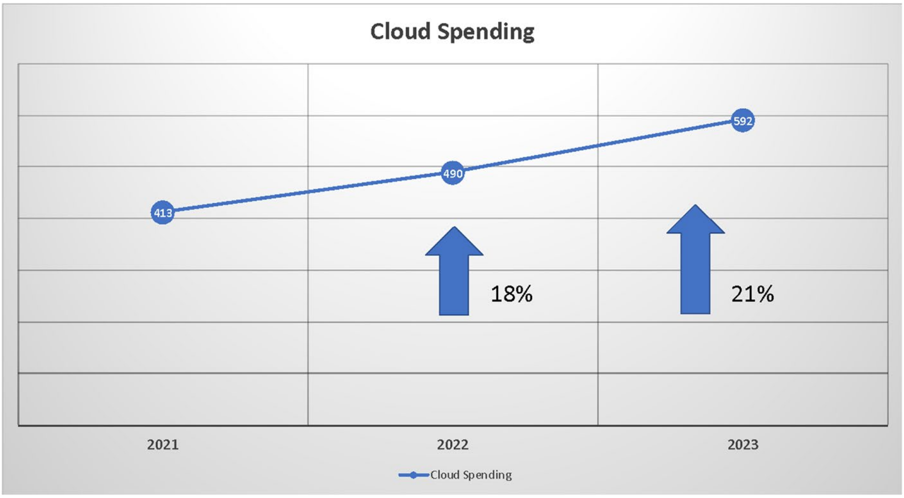

Figure 1-1. *Gartner's increase in cloud spending*

Gartner predicts (ref: https://www.gartner.com/en/newsroom/press-releases/2022-10-31-gartner-forecasts-worldwide-public-cloud-end-user-spending-to-reach-nearly-600-billion-in-2023) that by the end of 2023, cloud spending is likely to hit 592 billion USD. This is a 21 percent increase compared to the previous year's numbers at 490 billion USD. In fact, the year 2022 saw an 18 percent jump of cloud spending compared to 2021, which clocked in at 413 billion USD. These figures provide insights to those who haven't made the change yet. It is also important that leaders become cloud savvy so they can have meaningful conversations with their clients and teams.

The Evolution of the Cloud

There was a time when major organizations housed their own datacenters. These were secure locations that were access controlled, and in some places, a guard secured the premises. It was considered a privilege to enter these rooms. Once you entered, these massive rooms consisted of stacks of racks and a whole lot of computing devices sitting inside them. Datacenters like these are now a mere memory, with the racks and servers replaced by consoles and GUI-driven controls. This journey is a stamp of evolution in infrastructure technology, and more importantly, it sets the context for future technology growth predictions.

In fact, the cloud is not just about computing infrastructure in specific. The thick red line dividing hardware and software no longer exists, and infrastructure and software have become homogenous. The evolution to cloud computing is a saga that marches from datacenters to the cloud, and the software architecture has gone from monolithic devices to microservices.

Cloud usage can be explained in three stages, as shown in Figure 1-2:

- Pre-cloud era
- Infrastructure cloud era
- Holistic cloud era

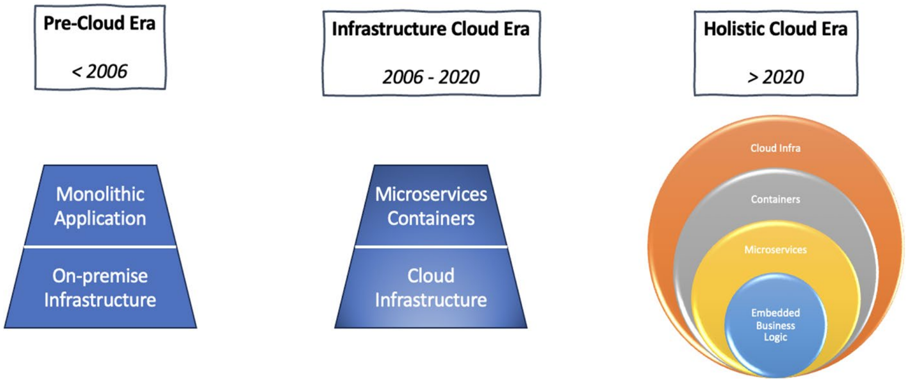

Figure 1-2. *Stages of cloud use*

Prior to 2006 (referred to as the *pre-cloud era*), there was no cloud. It was all datacenters and infrastructure hosted in datacenters. The applications were either monolithic or followed an *n*-tier architecture.

With the introduction of AWS, Azure, and GCP, organizations started to shift their infrastructure out of datacenters and into the cloud. Applications also shifted tack with the introduction of microservices, and container technology took shape. This was the transition era, with a number of organizations making the infrastructure switch to the cloud (hence, it was referred to as the infrastructure cloud era).

Today (the holistic cloud era), the cloud is a strategy for organizations to place all their eggs. There are multiple advantages to this, discussed later in this chapter. The business logic is integrated within the various cloud services, such as AI/ML, serverless, analytics, and so on, and applications no longer sit over the infrastructure. Instead, there is a harmony of various services working in tandem.

The Genesis of Networks

Before I cover computing infrastructure and software, it's important to discuss networks. The underlying technology that made cloud possible is the interconnecting network. Without the Internet, cloud computing would not have been possible. So, the evolution of cloud computing actually dates back to the 1960s, when the first networks were conceived. ARPANET (Advanced Research Projects Agency Network) was the first long-distance network developed in the United States. At the time, mainframes were expensive and mainframe users connected to them through dedicated systems connected to teletype machines. This was also during the cold war, and the thought of single points of failures if these computers were attacked was one of the main drivers for setting up remote networks.

ARPANET was implemented to connect University of California Los Angeles (UCLA) and the Stanford Research Institute (SRI). The mainframe was at SRI, while the users connected to it from UCLA. The network of systems grew through various universities and government institutions across the United States, and it spread to other parts of the world in the 1970s. ARPANET evolved and was replaced by the Internet in 1986. ARPANET was decommissioned in 1990.

Meanwhile, John McCarthy, the prominent American computer scientist, predicted during the MIT centennial in 1961 that "computing may someday be organized as a public utility just as the telephone system is a public utility." He also suggested that there would be a time when people would have to pay to use computers, which is a reality today.

Infrastructure Advancements

Accessing remote computers through networks was a major breakthrough for the evolution of computing, and the next transformation taking shape was the virtualization of infrastructure. Mainframes were expensive, and it was impractical to provision a mainframe for every user. Organizations allowed multiple users to utilize mainframes using the same access, which meant that the files and data were not compartmentalized but rather commonly accessible.

Virtualization

In the 1970s, mainframes got a major boost through the virtualization technology. Mainframe administrators could create multiple virtual machines (VM) on a single physical node. Each VM could run its own operating system, partition data, allocate CPU and memory, and set up other customizations. This VM operating system set the tone for the modern virtual machines, which have taken on a number of characteristics.

Server virtualization is a common feature that divides a physical server into multiple sub-servers. Each server created through virtualization comes with its own CPU, memory, and storage, among other computing resources. While IBM is credited for developing mainframe virtualization, VMware is the company that introduced virtualization in the 1990s to non-mainframe servers (the x86 architecture). X86 machines are typically the computers and servers that run on the Windows and Linux operating systems.

The virtualization technology leverages hypervisors that create a virtual platform on the physical node. This allows multiple instances of virtual machines to be created and used. In short, hypervisors manage the virtual machines on a physical server.

Virtualization using hypervisors flourished. Every datacenter around the globe used this technology. Hypervisor products like Microsoft's Hyper-V, VMware, and Citrix ruled the roost. Then Web 2.0 introduced a limitation.

Cloud Computing

As the digital industry grew, its users, applications, and the need for computing resources grew rapidly. The traditional virtualization was limited to a particular physical server. The maximum allocatable resources are constrained by the resources of the physical server. If a website hosted on a server demanded more RAM, the server had to be brought down, and physical RAM had to be added. With the rampant growth of digital services, this was no longer a feasible solution.

Cloud computing was the answer. It introduced scalability and reached beyond physical computing devices. The cloud technology that leveraged hypervisors could work across multiple physical nodes and pool together resources to present them as though they were coming from a single physical node. Think of it like combining all the servers in a datacenter to make it look like a single physical server. And if you want to build a server with lots of CPU power and RAM, the hypervisor can pull CPU and RAM from multiple systems to create a single powerful server. This upgraded hypervisor technology together with the spread of high-speed Internet made the cloud possible. The next level of flexibility was to access the hypervisors from a remote location that did not have dedicated lines. The only option was the Internet, and with the Internet on broadband, the cloud just eased its way in.

The idea of a cloud with a host of services was started by Amazon with their Amazon Web Services (AWS) in 2006. They started with Simple Storage Service (S3). The first remnants of the cloud were introduced by Salesforce back in 1999. They did this by providing their enterprise-grade software through a website. Users could subscribe to the services, and not really own the software. This concept is in vogue today and is called *Software as a Service* (SaaS).

Google entered the cloud space through their Google docs, which was based on their timely acquisition, Writely. Microsoft's Azure was launched in 2010, followed by IBM SmartCloud in 2011 (now called IBM Cloud).

The Software Evolution

When developers learned to code in those days, all the logic was embedded into a single application—be it how the data was transformed or how the transactions were processed. This was possibly the easiest and most straightforward way to develop an application, and this was the approach that was used in the professional software industry as well. Such applications were referred to as *monolithic applications*.

Monolithic applications are a single application with the all the logic embedded into a single process and componentized into multiple libraries. There is basically one single codebase for the business logic, the user interface, and the database calls, which is the root of all problems!

The problem is not the way the application is structured, but the ability to scale it. All the logic is embedded into a single process, so trying to make changes to one piece may break another piece. Since there is a single codebase, any change that is made, however small, requires the entire codebase to be compiled and tested. Any change risks breaking functionalities or killing performance due to the ripple effect. In other words, any change required the entire codebase to be rebuilt, tested, and deployed.

Think of this as a series of intricate wires running around a room, and you are asked to make changes to a few configurations. What is the probability that you get it right the first time? How much effort is needed to understand the layout of the wires and configure the new ones? What is the impact of wiring two wrong ends? These are the typical questions the change-management board poses whenever any changes are posted, and these questions unravel the complexity behind monolithic applications.

In short, monolithic applications are easy to build, but difficult to maintain. They are harder to scale, and reusing certain components of a monolithic application is the hardest bit.

Moving to *n*-tier

Single-tier architecture is a type of software architecture where all the components of the application are housed and run on a single platform. The software components include the presentation (the user interface), the application/business logic, and the data access/storage, and they reside on a single machine.

Single-tier architecture was predominant during the genesis of software development because it was simplistic and easy to deploy. The problem was that it was compatible with smaller applications and not preferred for applications that needed to be scaled and distributed.

As architectures evolved, the database in a single-tier was separated from the user interface and business logic. This led to a mini-revolution with multiple clients being able to access a single database and offer distributed services over the networks. This concept is called a *two-tier architecture*. Basically, the business logic and the user interface were encapsulated in a single tier, while the database was hosted on another tier.

The next step on the improvement ladder was to house the business logic and user interface into their own dedicated infrastructure. This provided further flexibility for developers to change the user interface without making changes to the business logic and vice versa. This is called a *three-tier architecture*.

The first tier is referred to as the presentation layer (the user interface). The user interface is the look and feel of a mobile application, web application, or a desktop application. This is followed by the application layer (the business logic), in which the information collected from the presentation layer is processed as per the required logic/configurations. It communicates with other layers through API calls. The application layer is developed in languages such as Python, Node.js, and Java, among others. The third layer is the data layer, where the database sits. It contains files and third-party database services. The application layer can read, modify, or delete data based on the logic and the information received from the presentation layer. The data layer typically consists of relational database such as Oracle, DB2 and MySQL, or NoSQL databases such as MongoDB and Cassandra.

Three-tier applications had a number of advantages over monolithic applications, mainly addressing the problems discussed in the previous section. As each of the tiers were physically detached from the others, development could happen on all three tiers simultaneously, which helped develop applications rapidly. Since the entire application was not compiled and deployed for every single change, the risks associated with impact to applications due to faulty changes was reduced greatly. Scaling too became simpler, as developers had to worry only about the tier in question rather than the entire application. Think of it like complexity divided by a third!

The Move to Service Oriented Architecture

The move from a monolithic to a three-tier application sparked a rush of further experimentation, and the result was the concept of building modular applications. These are not quite complete applications but rather small applications for a particular area. The concept is called *Service Oriented Architecture* (SOA) and it leverages software components called *services* that help develop business applications.

Think of a modular kitchen made of multiple cabinets, drawers, and shelves. Each of these cabinets, drawers, and shelves are independently constructed in a way so they work well together, in whatever configuration they are placed. The final outcome is a beautiful kitchen with all the customer's wants and needs taken care of. The manufacturer doesn't have to create individual elements separately, but rather they create a bulk product and retrofit it based on the customer's needs and desires. It's a win-win situation—the customer gets what they desire and at a reasonable price and the manufacturer can churn out more cabinets more quickly, leading to better profits.