SAMPLE CHAPTER

Google Cloud Platform IN ACTION

JJ Geewax

Foreword by Urs Hölzle

MANNING

Google Cloud Platform in Action

by JJ Geewax

Chapter 3

brief contents

3 The cloud data center

This chapter covers

- What data centers are and where they are
- Data center security and privacy
- Regions, zones, and disaster isolation

If you've ever paid for web hosting before, it's likely that the computer running as your web host was physically located in a data center. As you learned in chapter 1, deploying in the cloud is similar to traditional hosting, so, as you'd expect, if you turn on a virtual machine in, or upload a file to, the cloud, your resources live inside a data center. But where are these data centers? Are they safe? Should you trust the employees who take care of them? Couldn't someone steal your data or the source code to your killer app?

All of these questions are valid, and their answers are pretty important—after all, if the data center was in somebody's basement, you might not want to put your banking details on that server. The goal of this chapter is to explain how data centers have evolved over time and highlight some of the details of Google Cloud Platform's data centers. Google's data centers are pretty impressive (as shown in figure 3.1), but this isn't a fashion show. Before you decide to run mission-critical stuff in a data center, you probably want to understand a little about how it works.

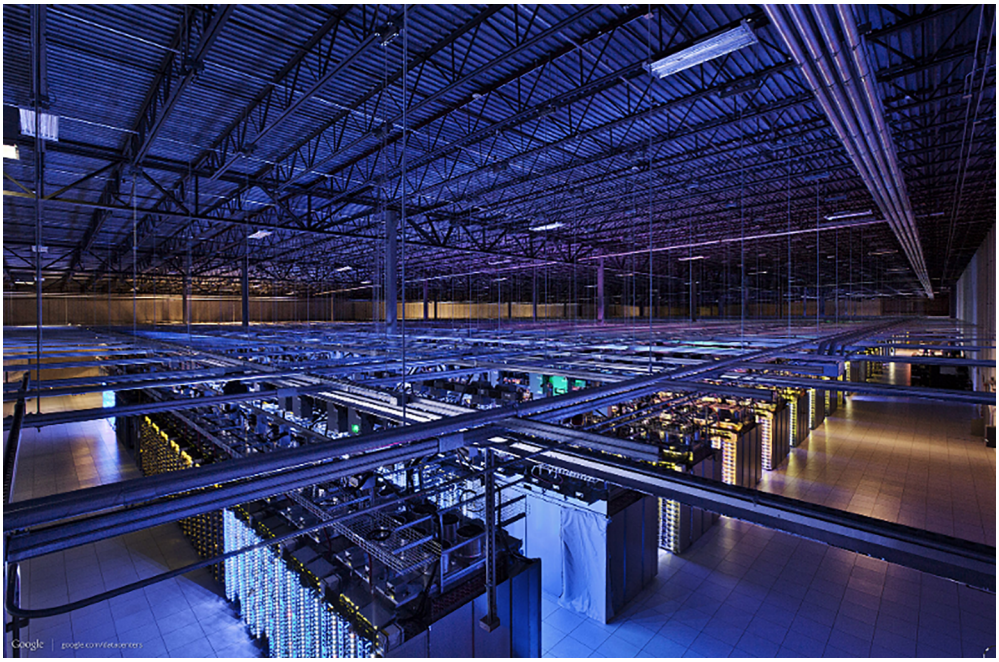

Figure 3.1 A Google data center

Keep in mind that many of the things you'll read in this chapter about data centers are industrywide standards, so if something seems like a great feature (such as strict security to enter the premises), it probably exists with other cloud providers as well (like Amazon Web Services or Microsoft Azure). I'll make sure to call out things that are Google-specific so it's clear when you should take note. I'll start by laying out a map to understand Google Cloud's data centers.

3.1 Data center locations

You might be thinking that *location* in the world of the cloud seems a bit oxymoronic, right? Unfortunately, this is one of the side effects of marketers pushing the cloud as some amorphic mystery, where all of your resources are multihomed rather than living in a single place. As you'll read later, some services do abstract away the idea of location so that your resources live in multiple places simultaneously, but for many services (such as Compute Engine), resources live in a single place. This means you'll likely want to choose one near your customers.

To choose the right place, you first need to know what your choices are. As of this writing, Google Cloud operates data centers in 15 different regions around the world, including in parts of the United States, Brazil, Western Europe, India, East Asia, and Australia. See figure 3.2.

Figure 3.2 Cities where Google Cloud has data centers and how many in each city (white balloons indicate "on the way" at the time of this writing.)

This might not seem like a lot, but keep in mind that each city has many different data centers for you to choose from. Table 3.1 shows the physical places where your data resources can exist.

Table 3.1 Zone overview for Google Cloud

Region	Location	Number of data centers
Total		44
Eastern US	South Carolina, USA	3
Eastern US	North Virginia, USA	3
Central US	Iowa, USA	4
Western US	Oregon, USA	3
Canada	Montréal, Canada	3
South America	São Paulo, Brazil	3
Western Europe	London, UK	3
Western Europe	Belgium	3
Western Europe	Frankfurt, Germany	3
Western Europe	Netherlands	2
South Asia	Mumbai, India	3
South East Asia	Singapore	2
East Asia	Taiwan	3

Table 3.1 Zone overview for Google Cloud *(continued)*

Region	Location	Number of data centers
North East Asia	Tokyo, Japan	3
Australia	Sydney, Australia	3

How does this stack up to other cloud providers, as well as traditional hosting providers? Table 3.2 will give you an idea.

Table 3.2 Data center offerings by provider

Provider	Data centers
Google Cloud	44 (across 15 cities)
Amazon Web Services	49 (across 18 cities)
Azure	36 (across 19 cities)
Digital Ocean	11 (across 7 cities)
Rackspace	6

Looking at these numbers, it seems that Google Cloud is performing pretty well compared to the other cloud service providers. That said, two factors might make you choose a provider based on the data center locations it offers, and both are focused on network latency:

- You need ultralow latency between your servers and your customers. An example here is high-frequency trading, where you typically need to host services only microseconds away from a stock exchange, because responding even one millisecond slower than your competitors means you'll lose out on a trade.
- You have customers that are far away from the nearest data center. A common example is businesses in Australia, where the nearest options for some services might still be far away. This means that even something as simple as loading a web page from Australia could be frustratingly slow.

NOTE I cover a third reason based on legal concerns in section 3.3.3.

If your requirements are less strict, the locations of data centers shouldn't make too much of a difference in choosing a cloud provider. Still, it's important to understand your latency requirements and how geographical location might affect whether you meet them or not (figure 3.3).

Now that you know a bit about where Google Cloud's data centers are and why location matters, let's briefly discuss the various levels of isolation. You'll need to know about them to design a system that will degrade gracefully in the event of a catastrophe.

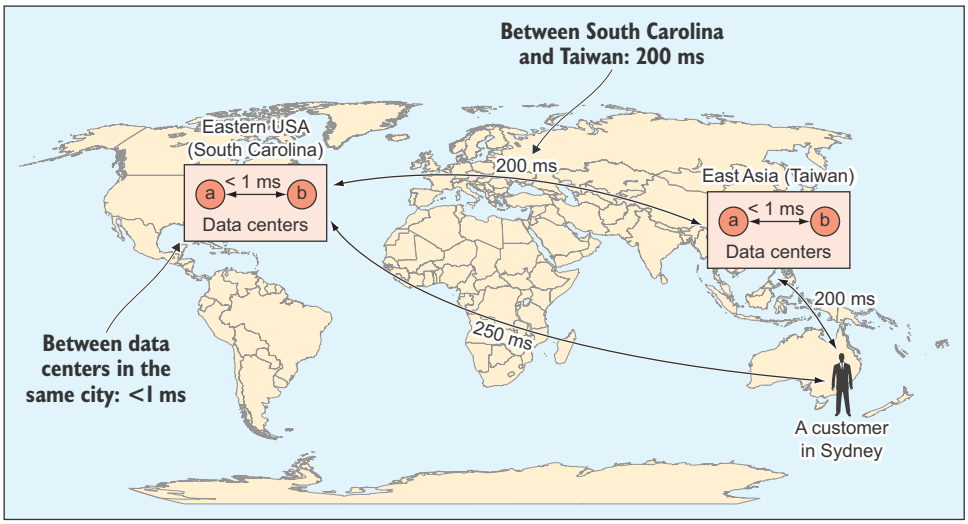

Figure 3.3 Latencies between different cities and data centers

3.2 Isolation levels and fault tolerance

Although I've talked about cities, regions, and data centers, I haven't defined them in much detail. Let's start by talking about the types of places where resources can exist.

3.2.1 Zones

A *zone* is the smallest unit in which a resource can exist. Sometimes it's easiest to think of this as a single facility that holds lots of computers (like a single data center). This means that if you turn on two resources in the same zone, you can think of that as the two resources living not only geographically nearby, but in the same physical building. At times, a single zone may be a bunch of buildings, but the point is that from a latency perspective (the `ping` time, for example) the two resources are close together.

This also means that if some natural disaster occurs—maybe a tornado comes through town—resources in this single zone are likely to go offline together, because it's not likely that the tornado will take down only half of a building, leaving the other half untouched. More importantly, it means that if a malfunction such as a power outage occurred, it likely would affect the entire zone. In the various APIs that take a `zone` (or `location`) as a parameter, you'll be expected to specify a zone ID, which is a unique identifier for a particular facility and looks something like `us-east1-b`.

3.2.2 Regions

Moving up the stack, a collection of zones is called a *region*, and this corresponds loosely to a city (as you saw in table 3.1), such as Council Bluffs, Iowa, USA. If you turn on two resources in the same region but different zones, say `us-east1-b` and `us-east1-c`, the resources will be somewhat close together (meaning the latency between them will be

shorter than if one resource were in a zone in Asia), but they're guaranteed to not be in the same physical facility.

In this case, although your two resources might be isolated from zone-specific failures (like a power outage), they might not be isolated from catastrophes (like a tornado). See figure 3.4. You might see regions abbreviated by dropping the last letter on the zone. For example, if the zone is `us-central1-a`, the region would be `us-central1`.

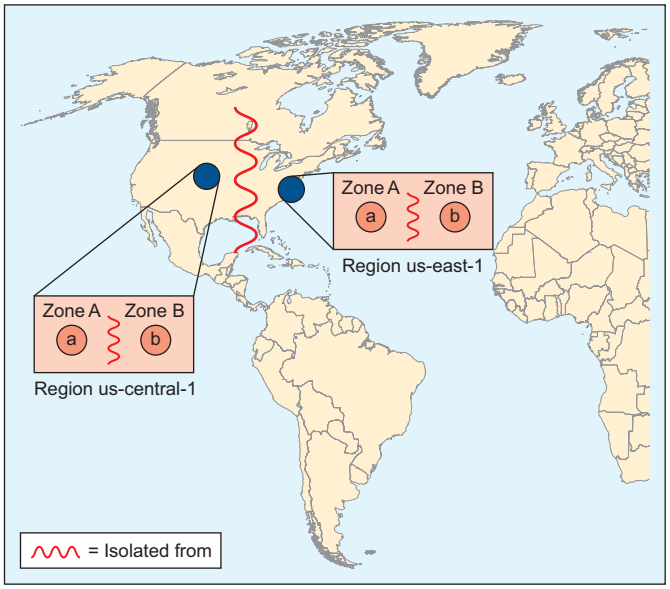

Figure 3.4 A comparison of regions and zones

3.2.3 Designing for fault tolerance

Now that you understand what zones and regions are, I can talk more specifically about the different levels of isolation that Google Cloud offers. You might also hear these described as *control planes*, borrowing the term from the networking world. When I refer to isolation level or the types of control plane, I'm talking about what thing would have to go down to take your service down with it. Services are available, and can be affected, at several different levels:

- *Zonal*—As I mentioned in the example, a service that's *zonal* means that if the zone it lives in goes down, it also goes down. This happens to be both the easiest type of service to build—all you need to do is turn on a single VM and you have a zonal service—and the least highly available.
- *Regional*—A regional service refers to something that's replicated throughout multiple zones in a single region. For example, if you have a MongoDB instance living in `us-east1-b`, and a hot-failover living in `us-east1-c`, you have a regional service. If one zone goes down, you automatically flip to the instance in the other zone. But if an earthquake swallows the entire city, both zones will

go down with the region, taking your service with it. Although this is unlikely, and regional services are much less likely to suffer outages, the fact that they're geographically colocated means you likely don't have enough redundancy for a mission-critical system.

- *Multiregional*—A multiregional service is a composition of several different regional services. If some sort of catastrophe occurs that takes down an entire region, your service should still continue to run with minimal downtime (figure 3.5).
- *Global*—A global service is a special case of a multiregional service. With a global service, you typically have deployments in multiple regions, but these regions are spread around the world, crossing legal jurisdictions and network providers. At this point, you typically want to use multiple cloud providers (for example, Amazon Web Services alongside Google Cloud) to protect the service against disasters spanning an entire company.

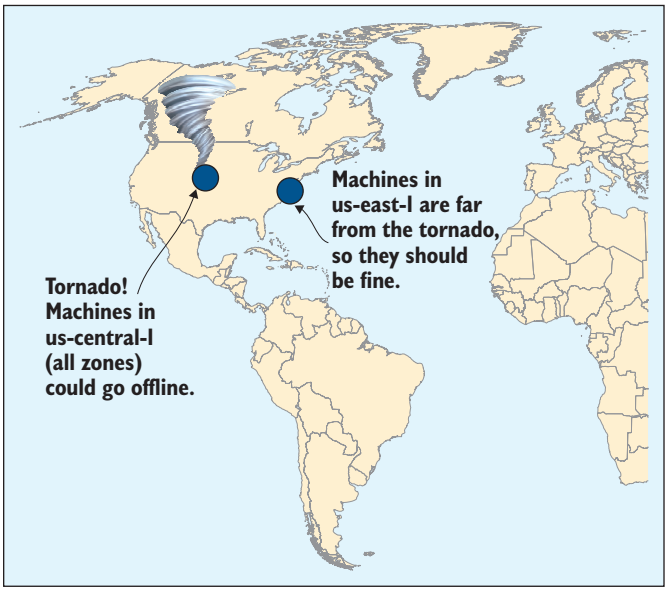

Figure 3.5 Disasters like tornadoes are likely to affect a single region at a time.

For most applications, regional or even zonal configurations will be secure enough. But as you become more mission-critical to your customers, you'll likely start to consider more fault-tolerant configurations, such as multiregional or global.

The important thing when building your service isn't primarily using the most highly available configuration, but knowing what your levels of fault tolerance and isolation are at any time. Armed with that knowledge, if any part of your system becomes absolutely critical, you at least know which pieces will need redundant deployments and where those new resources should go. I'll talk much more about redundancy and high availability when I discuss Compute Engine in chapter 9.

3.2.4 Automatic high availability

Over the years, certain common patterns have emerged that show where systems need to be highly available. Based on these patterns, many cloud providers have designed richer systems that are automatically highly available. This means that instead of having to design and build a multiregional storage system yourself, you can rely on Google Cloud Storage, which provides the same level of fault isolation (among other things) for your basic storage needs.

Several other systems follow this pattern, such as Google Cloud Datastore, which is a multiregional nonrelational storage system that stores your data in five different zones, and Google App Engine, which offers two multiregional deployment options (one for the United States and another for Europe) for your computing needs. If you run an App Engine application, save some data in Google Cloud Storage, or store records in Google Cloud Datastore, and an entire region explodes, taking down all zones with it, your application, data, and records all will be fine and remain accessible to you and your customers. Pretty crazy, right?

The downside of products like these is that typically you have to build things with a bit more structure. For example, when storing data on Google Cloud Datastore, you have to design your data model in a way that forces you to choose whether you want queries to always return the freshest data, or you want your system to be able to scale to large numbers of queries.

You can read more about this in the next few chapters, but it's important to know that although some services will require you to build your own highly available systems, others can do this for you, assuming you can manage under the restrictions they impose. Now that you understand fault tolerance, regions, zones, and all those other fun things, it's time to talk about a question that's simple yet important, and sometimes scary: Is your stuff safe?

3.3 Safety concerns

Over the past few years, personal and business privacy have become a mainstream topic of conversation, and for good reason. The many leaks of passwords, credit card data, and personal information have led the online world to become far less trusting than it was in the past. Customers are now warier of handing out things like credit card numbers or personal information. They're legitimately afraid that the company holding that information will get hacked or a government organization will request access to the data under the latest laws to fight terrorism and increase national security. Put bluntly, putting your servers in someone else's data center typically involves giving up some control over your assets (such as data or source code) in exchange for other benefits (such as flexibility or lower costs). What does this mean for you? A good way to understand these trade-offs is to walk through them one at a time. Let's start with the security of your resources.

3.3.1 Security

As you learned earlier, when you store data or turn on a computer using a cloud provider, although it's marketed as living nowhere in particular, your resources do physically exist somewhere, sometimes in more than one place. The biggest question for most people is … where?

If you store a photo on a hard drive in your home, you know exactly where the photo is—on your desk. Alternatively, if you upload a photo to a cloud service like Google Cloud Storage or Amazon's S3, the exact location of the data is a bit more complicated to determine, but you can at least pinpoint the region of the world where it lives. On the other hand, the entire photo is unlikely to live in only one place—different pieces of multiple copies of the file likely are stored on lots of disk drives. What do you get for this trade-off? Is more ambiguity worth it? When you use a cloud service to do something like store your photos, you're paying for quite a bit more than the disk space; otherwise, the fee would be a flat rate per byte rather than a recurring monthly fee.

To understand this in more detail, let's look at a real-life example of storing a photo on a local hard drive. By thinking about all the things that can go wrong, you can start to see how much work goes into preventing these issues and why the solution results in some ambiguity about where things exist. After we go through all of these things, you should understand how exactly Google Cloud prevents them from happening and have some more clarity regarding what you get by using a cloud service instead of your own hard drive.

When talking about securing resources, you typically have three goals:

- *Privacy*—Only authorized people should be able to access the resources.
- *Availability*—The resources should never be inaccessible to authorized people.
- *Durability*—The resources should never be corrupted or go missing.

In more specific terms with you and your photo, that would be

- *Privacy*—No one besides you should be able to look at your photo.
- *Availability*—You should never be told "Not right now, try again later!" when you ask to look at your photo.
- *Durability*—You should never come back and find your photo deleted or corrupted.

The goals seem simple enough, right? Let's look at how this breaks down with your hard drive at home when real life happens, so to speak. The first thing that can go wrong is simple theft. For example, if someone breaks into your home and steals your hard drive, the photo you stored on that drive is now gone. This breaks your goals for availability and durability right off the bat. If your photo wasn't encrypted at all, this also breaks the privacy goal, as the thief can now look at your photo when you don't want anyone else to do so.

You can lump the next thing that can go wrong into a large group called unexpected disasters. This includes natural disasters, such as earthquakes, fires, and floods,

but in the case of storing data at home, it also includes more common accidents, such as power surges, hard drive failures, and kids spilling water on electronic equipment.

After that, you have to worry about more nuanced accidents, such as accidentally formatting the drive because you thought it was a different drive or overwriting files that happened to have similar names. These issues are more complicated because the system is doing as it was told, but you're accidentally telling it to do the wrong thing. Finally, you have to worry about network security. If you expose your system on the internet and happen to use a weak password, it's possible that an intruder could gain access to your system and access your photo, even if you encrypted the photo.

All of these types of accidents break the availability and durability goals, and some of them break the privacy goals. So how do cloud providers plan for these problems? Couldn't you do this yourself? The typical way cloud providers deal with these problems comes down to a few tactics:

- *Secure facilities*—Any facility housing resources (like hard drives) should be a high-security area, limiting who can come and go and what they can take with them. This is to prevent theft as well as sabotage.
- *Encryption*—Anything stored on disks should be encrypted. This is to prevent theft compromising data privacy.
- *Replication*—Data should be duplicated in many different places. This is to prevent a single failure resulting in lost data (durability) as well as a network outage limiting access to data (availability). This also means that a catastrophe (such as a fire) would only affect one of many copies of the data.
- *Backup*—Data should be backed up off-site and can be easily restored on request. This is to prevent a software bug accidentally overwriting all copies of the data. If this happens, you could ask for the old (correct) copy and disregard the new (erroneous) copy.

As you might guess, providing this sort of protection in your own home isn't just challenging and expensive—by definition it requires you to have more than one home! Not only would you need advanced security systems, you'd need full-time security guards, multiple network connections to each of your homes, systems that automatically duplicated data across multiple hard drives, key management systems for storing your encryption keys, and backups of data on rolling windows to different locations. I can comfortably say that this isn't something I'd want to do myself. Suddenly, a few cents per gigabyte per month doesn't sound all that bad.

3.3.2 Privacy

What about the privacy of your data? Google Cloud Storage might keep your photo in an encrypted form, but when you ask for it back, it arrives unencrypted. How can that be? The truth here is that although data is stored in encrypted form and transferred between data centers similarly, when you ask for your data, Google Cloud does have

the encryption key and uses it when you ask for your photo. This also means that if Google were to receive a court order, it does have the technical ability to comply with the order and decrypt your data without your consent.

To provide added security, many cloud services provide the ability to use your own encryption keys, meaning that the best Google can do is hand over encrypted data, because it doesn't have the keys to decrypt it. If you're interested in more details about this topic, you can learn more in chapter 8, where I discuss Google Cloud Storage.

3.3.3 Special cases

Sometimes special situations require heightened levels of security or privacy; for example:

- Government agencies often have strict requirements.
- Companies in the U.S. healthcare industry must comply with HIPAA regulations.
- Companies dealing with the personal data of German citizens must comply with the German BDSG.

For these cases, cloud providers have come up with a few options:

- Amazon offers GovCloud to allow government agencies to use AWS.
- Google, Azure, and AWS will all sign BAAs to support HIPAA-covered customers.
- Azure and Amazon offer data centers in Germany to comply with BDSG.

Each of these cases can be quite nuanced, so if you're in one of these situations, you should know

- It's still possible to use cloud hosting.
- You may be slightly limited as to which services you can use.

You're probably best off involving legal counsel when making these kinds of serious decisions about hosting providers. All that said, hopefully you're now relatively convinced that cloud data centers are safe enough for your typical needs, and you're open to exploring them for your special needs. But I still haven't touched on the idea of sharing these data centers with all the other people out there. How does that work?

3.4 Resource isolation and performance

The big breakthrough that opened the door to cloud computing was the concept of virtualization, or breaking a single physical computer into smaller pieces, each one able to act like a computer of its own. What made cloud computing amazing was the fact that you could build a large cluster of physical computers, then lease out smaller virtual ones by the hour. This process would be profitable as long as the leases of the smaller virtual computers covered the average cost to run the physical computers.

This concept is fascinating, but it omits one important thing: Do two virtual half computers run as fast as one physical whole computer? This leads to further

questions, such as whether one person using a virtual half computer could run a CPU-intensive workload that spills over into the resources of another person using a second virtual half computer and effectively steal some of the CPU cycles from the other person. What about network bandwidth? Or memory? Or disk access? This issue has come to be known as the noisy neighbor problem (figure 3.6) and is something everyone running inside a cloud data center should understand, even if superficially.

Figure 3.6 Noisy neighbors can impinge on those nearby.

The short answer to those questions is that you'll only get perfect resource isolation on *bare metal* (nonvirtualized) machines.

Luckily, many of the cloud providers today have known about this problem for quite a long time and have spent years building solutions to it. Although there's likely no perfect solution, many of the preventative measures can be quite good, to the point where fluctuations in performance might not even be noticeable.

In Google's case, all of the cloud services ultimately run on top of a system called Borg, which, as you can read in *Wired* magazine from March 2013, "is a way of efficiently parceling work across Google's vast fleet of … servers." Because Google uses the same system internally for other services (such as Gmail and YouTube), resource isolation (or perhaps better phrased as *resource fairness*) is a feature that has almost a decade of work behind it and is constantly improving. More concretely, for you this means that if you purchase 1 vCPU worth of capacity on Google Compute Engine, you should get the same number of computing cycles, regardless of how much work other VMs are trying to do.

Summary

- Google Cloud has many data centers in lots of locations around the world for you to choose from.
- The speed of light is the limiting factor in latency between data centers, so consider that distance when choosing where to run your workloads.
- When designing for high availability, always use multiple zones to avoid zone-level failures, and if possible multiple regions to avoid regional failures.
- Google's data centers are incredibly secure, and its services encrypt data before storing it.
- If you have special legal issues to consider (HIPAA, BDSG, and so on), check with a lawyer before storing information with any cloud provider.

CLOUD

Google Cloud Platform IN ACTION

JJ Geewax

Thousands of developers worldwide trust Google Cloud Platform, and for good reason. With GCP, you can host your applications on the same infrastructure that powers Search, Maps, and the other Google tools you use daily. You get rock-solid reliability, an incredible array of prebuilt services, and a cost-effective, pay-only-for-what-you-use model. This book gets you started.

Google Cloud Platform in Action teaches you how to deploy scalable cloud applications on GCP. Author and Google software engineer JJ Geewax is your guide as you try everything from hosting a simple WordPress web app to commanding cloud-based AI services for computer vision and natural language processing. Along the way, you'll discover how to maximize cloud-based data storage, roll out serverless applications with Cloud Functions, and manage containers with Kubernetes. Broad, deep, and complete, this authoritative book has everything you need.

What's Inside

- The many varieties of cloud storage and computing
- How to make cost-effective choices
- Hands-on code examples
- Cloud-based machine learning

Written for intermediate developers. No prior cloud or GCP experience required.

JJ Geewax is a software engineer at Google, focusing on Google Cloud Platform and API design.

To download their free eBook in PDF, ePub, and Kindle formats, owners of this book should visit manning.com/books/google-cloud-platform-in-action

"Demonstrates how to use GCP in practice while also explaining how things work under the hood."

—From the Foreword by Urs Hölzle, SVP, Technical Infrastructure, Google

"Provides powerful insight into Google Cloud, with great worked examples."

—Max Hemingway
DXC Technology

"A great asset when migrating to Google Cloud, not only for developers, but for architects and management too."

—Michał Ambroziewicz, Netsprint

"As an Azure user, I got great insights into Google Cloud and a comparison of both providers. A must-read."

—Grzegorz Bernas
Antaris Consulting

ISBN-13: 978-1-61729-352-8
ISBN-10: 1-61729-352-0
55999
9 781617 293528

MANNING $59.99 / Can $79.99 [INCLUDING eBOOK]