THE EXPERT'S VOICE® IN CLOUD COMPUTING

Building Your Next Big Thing with Google Cloud Platform

A Guide for Developers and Enterprise Architects

S. P. T. Krishnan and Jose L. Ugia Gonzalez

Apress®

For your convenience Apress has placed some of the front matter material after the index. Please use the Bookmarks and Contents at a Glance links to access them.

Apress®

Contents at a Glance

Introduction

Cloud computing, specifically the public cloud, is revolutionizing the way application developers design, build, deploy, maintain, and retire their software. Just a decade ago, it took several weeks to make a website public. Today, thanks to public cloud platforms like Amazon Web Services, Google Cloud Platform, and Microsoft Azure, the same task can be done in a hour, if not in a couple of minutes.

When we became Google Developer Experts in Google Cloud Platform, we interacted with the developer communities in several countries and asked them what they needed to in order to start using Cloud Platform. The unanimous responses were the need for books. We scoured the market, and although a few books talked about specific Cloud Platform technologies, we couldn't find a single book that introduced application developers to the entire Cloud Platform. So, we decided to fill the gap—and the result is this book.

We started writing with one clear objective: to help you benefit from the power of Cloud Platform to make an impact on your present and future projects at work, on the side, in your hobbies, or in any other area where taking advantage of the experience acquired by Google in recent years can get you further and faster than before.

Let's step back for a second and see how technological progress has affected the way you work. Think about a day of work in your life, starting with small things like commuting, organizing meetings, managing productive time, and so on. The important point is not how much these have changed but the fact that a few years ago we never would have expected to be working with the Internet or reading an ebook on our way to work; meeting with colleagues in different parts of the world in a productive way; or controlling our work habits, focus times, and breaks with tools that you can install on your computer. We did not see many of these things coming; and even when we did, we tended not to accept them until they penetrated our culture sufficiently that not adopting them would have left us behind.

Because of the pace at which technology progresses, this process repeats itself every few years. So regardless of how new you are to technology, it is likely that you have seen this cycle a couple of times already. It does not matter how many times this happens—most of us are static and defensive in the face of change, because it is easier to think in retrospective than to apply broad new knowledge to our daily lives. If we did, it would be clear to us that in the near future, information will surround us in less invasive ways than it does today when we use computers or mobile devices. We would also know that artificial intelligence and machine learning will likely keep handling more duties for humans; and that our lives will be connected not only to other lives, but also to the objects that surround us—houses, cars, streets, buildings—and so on. Likewise, and most important, we know that developing server applications, in most cases, will not require us to set up machines, databases, and load balancers—at least, not by ourselves. If we need to analyze and process big chunks of information, we will not need to set up the entire infrastructure; or if we need massive amounts of computing power to make calculations that are still out of reach today, we will be ready to run the logic in a matter of seconds.

This book is intended to help you make that transition in Cloud Platform and build a foundation that will make you comfortable in such a flexible and changing environment. You can consume this book in two different ways. You can read it the way you read most books, starting with chapter one and reading all the way to the end. If you do, you will get a broad and experimental understanding of the entire stack of services that Cloud Platform offers. This will give you the assets you need to design and tackle today's challenges when it comes to cloud computing.

Conversely, you can use this book as a travel companion through your ideas, projects, or work, jumping between chapters based on your needs at specific points in time. For example, suppose you decide to start gathering and processing analytics in your company. You can open Chapter 10 of this book, learn about Google BigQuery, and get your system set up and ready in a few pages. Or consider a different project: you want to build something very fast in order to get your product or service out as soon as possible. In that case, you can jump directly to Chapter 5, where we cover Google App Engine, or Chapter 14, about Google Cloud Endpoints, and get your back end set up in a matter of hours. Don't worry; when we think it is relevant for you to read about other technologies, we point you to the right resources inside and outside of this book.

Who This Book Is For

This book is targeted at two classes of developers: those new to cloud computing and those new to Cloud Platform. We take an on-ramp approach and gradually introduce you first to cloud computing and the public cloud and then to Cloud Platform. We adopt a "getting things done" approach (versus a "tell-all" approach) and share only essential knowledge that is required for you to get going with Cloud Platform.

Downloading the Code

The source code for the examples in this book can be downloaded from `github.com/googlecloudplatformbook`, and the errata will be posted at `www.cloudplatformbook.com`. The source code for this book is available in zip file format at `www.apress.com/9781484210055`.

Contacting the Authors

The authors can be reached at `cloudplatformbook@gmail.com`.

PART I

Introducing Cloud Computing and Google Cloud Platform

CHAPTER 1

The Google Cloud Platform Difference

Cloud computing as a vision is just 54 years young in 2015 (much older than either of this book's authors!). In 1961, John McCarthy introduced the idea of "computation being delivered as a public utility." Over the next five decades, various technological innovations enabled today's cloud computing, including the following:

- In 1960s, J. C. R. Licklider developed ARPANET—the forerunner to the Internet and what is considered to be the biggest contributor to the history of cloud computing in this era.
- In 1971, Intel engineer Ray Tomlinson developed software that allowed users to send messages from one computer to another. This subsequently was recognized as the first e-mail.
- In 1976, Xerox's Robert Metcalfe introduced Ethernet, essentially standardizing the wired network interface in computers.
- In 1991, CERN released the World Wide Web for general (that is, noncommercial) use.
- In 1993, the Mosaic web browser allowed graphics to be shown on the Internet. In the same year, private companies were allowed to use the Internet for the first time.
- During the late 1990s and early 2000s (famously known as the dot-com era), the availability of multitenant architectures, widespread high-speed bandwidth, and global software interoperability standards created the right environment for cloud computing to finally take off.

The realization of a global high-speed network and a utilities-based business model are the two major driving principles behind cloud computing.

What Is Cloud Computing?

Cloud computing is about abstracting the computing infrastructure and other associated resources and offering them as service, usually on a pay-per-use basis, over the Internet. The service can be targeted for human consumption or consumption by other software systems. Users just need a web browser to access services; software systems can consume services using a web application programming interface (API). This abstraction is often realized through a technical process called *virtualization*.

WHAT IS VIRTUALIZATION?

Virtualization is a process through which a hardware resource (such as a server or network) is cloned as an in-memory resource and is used as the (virtual) foundation to support a software stack. Virtualization is not an entirely new concept; virtual memory, for example, is used extensively in modern operating system(s) for security, for process isolation, and to create an impression that more memory is available than is actually present. Virtualization also makes it easy to transfer a virtual resource to another system when the underlying hardware fails.

A good analogy to cloud computing is the electric grid that centralized the production, transmission, and distribution of electricity to consumers. Consumers simply plug in to the grid, consume power, and pay for what they use without worrying about the nitty-gritty details of how electricity is produced, transmitted, and distributed. (You may be interested to know that, before the electric grid was invented, each organization produced its own electricity. Obviously, this required a large capital expense and was affordable only for the elite and rich.)

Cloud technology standardizes and pools IT resources and automates many of the maintenance tasks done manually today. Cloud architectures facilitate elastic consumption, self-service, and pay-as-you-go pricing. *Cloud* in this context refers to cloud computing architecture, encompassing both public and private clouds. But the public cloud has its own distinct set of advantages, which are hard to replicate in a private setting. This chapter focuses on these from both technical and nontechnical perspectives.

Technical Benefits of Using a Public Cloud

Several key performance benefits may motivate you to migrate to the public cloud. This section covers a few of these benefits.

Uptime

Most public cloud providers have redundancy built in as part of their system design. This extends from foundational utilities like electricity, Internet, and air conditioning to hardware, software, and networking. As a result, providers typically can offer uptime of 99.9% or more. This translates to expected downtime of just 8.76 hours per year (~1/3 day). All businesses can benefit from such high uptime for their IT infrastructure.

As independent businesses, public cloud service providers are able to provide legally binding service-level agreements (SLAs) that state the guaranteed uptime for their infrastructure and the penalties when those guarantees are not met. Such SLAs are not typically available from internal IT departments. The following URLs are for the SLAs of some of the popular cloud platform products covered in this book. In general, once a product is out of beta and into general availability (GA), the corresponding SLA should be available at `https://cloud.google.com/<product>/sla`:

- `https://cloud.google.com/compute/sla`
- `https://cloud.google.com/appengine/sla`
- `https://cloud.google.com/sql/sla`
- `https://cloud.google.com/storage/sla`
- `https://cloud.google.com/datastore/sla`
- `https://cloud.google.com/bigquery/sla`

Resource Utilization

Many organizational applications' resource needs vary by time. (Here, *resource* is a generic term and may refer to CPU, RAM, disk traffic, or network traffic.) As an example, an employee-facing app may be used more during the day and require more resources; it uses fewer resources at night due to reduced demand. This time-of-day variability leads to low overall resource usage in a traditional data-center setup. When you use a public cloud infrastructure, more resources can be (instantly) deployed when required and released when not needed, leading to cost savings.

Public cloud service providers have wide visibility on resource usage patterns across their customers and typically cluster them based on industry. Any application's resource usage may vary across individual system components; this is known as *multi-resource variability*. Resource usage patterns across industries are known as *industry-specific variability*.

Due to resource usage visibility, a public cloud service provider can reassign resources released by one customer to another customer, thereby keeping resource utilization high. If there is no demand for a particular resource, the provider may shut down the corresponding infrastructure to save operational costs. This way, the provider is able to handle applications whose resource needs are spiky in nature.

Expertise

Pubic cloud service providers have experienced system and network administrators along with 24×7 hardware maintenance personnel on site, owing to the tight SLAs they provide. By using a public cloud, companies can indirectly tap on this expert pool.

It would be challenging for a small or medium-size business to recruit, train, and maintain a top-notch team of domain experts, especially when deployment size is limited. Even larger companies are sometimes unable to match the deep expertise available at a public cloud service provider. For example, the well-known file-sharing company DropBox, which has millions of users, runs entirely on a public cloud.

Economic Benefits of Using a Public Cloud

In addition to the technical benefits of using a public cloud, there are several economic advantages to doing so. This section discusses the economic benefits of deploying on a public cloud, based on typical business yardsticks.

TCO

Total cost of ownership (TCO) refers to the total cost of acquiring, using, maintaining, and retiring a product. When you understand TCO, you will realize that many hidden costs usually are not accounted for. Specifically, TCO should include core costs such as the actual price of hardware/software and non-core costs such as time spent on pre-purchase research, operating costs including utilities, manpower, maintenance, and so on. Non-core costs typically are not included with traditional purchases and are bundled into administrative costs.

In the context of public cloud computing, TCO usually refers to software and/or hardware made available via lease. Interestingly, it avoids many non-core costs such as purchase-order processing, shipping, installation and so on.

Economies of Scale

Businesses (or customers) save more when they make a bulk purchase—the seller is willing to reduce its profit margin per unit for large sales. This is how big buyers, such as large companies, are able to get better deals compared to smaller companies in traditional transactions.

In the case of a public cloud, the buyer is the public cloud service provider such a Google Cloud Platform or Amazon Web Services. The larger the public cloud service provider, the more hardware it is likely to purchase from OEMs and the lower the price per unit. Public cloud service providers typically pass some of these savings to their customers (similar to a cooperative society model). This practice puts individual developers and companies of all sizes on the same level playing field, because they get the same low pricing for hardware/software.

CapEx and OpEx

Capital expenditures (CapEx) and *operational expenditures (OpEx)* are linked and refer to expenses incurred at different points in a product's consumption lifecycle. CapEx usually refers to large upfront expenses incurred before commencing use of a product, such as building a data center or acquiring hardware such as servers and racks and procuring Internet connectivity. OpEx refers to the associated operational expenses after a product is purchased and during its lifetime, such as manpower, utilities, and maintenance. The traditional wisdom is that high CapEx leads to low OpEx, whereas low CapEx leads to higher OpEx. Largely due to economies of scale, a public cloud service consumer enjoys low CapEx and low OpEx while transferring the large CapEx to the public cloud service provider, essentially creating a new economic model.

ROI and Profit Margins

Return on investment (ROI) and *profit margins* are strongly linked to one another and are key selling points for adopting a public cloud. ROI refers to the financial gain (or return) on an investment, and the profit margin is the ratio of income to revenue. By using a public cloud, an organization reduces its expenditures, and thus its ROI and profit margins are higher. Such higher returns are more visible in small and medium-sized businesses that have relatively high CapEx (because of low purchase quantities) when starting up.

Business Benefits of Using a Public Cloud

In addition to the technical and economic benefits, there are several business-process advantages to using a public cloud. This section describes a few of them.

Time to Market

Responsiveness is crucial in today's business environment. Business opportunities often arrive unannounced and are short-lived. Winners and losers are often determined by who is able to move faster and grab opportunities. Such opportunities typically require new/additional IT resources, such as computational power or bandwidth. A cloud service provider can provide these almost instantaneously. Hence, by using a public cloud, any business can reduce the time it takes to bring a product to market. In comparison, using the traditional route of building/acquiring infrastructure first, introducing a new product would require days if not weeks of onsite deployment.

Using a public cloud leads to reduced opportunity costs, increases agility, and makes it easy to respond to new opportunities and threats. The same quick response times also apply to shedding unneeded capacity. In summary, public cloud computing enables just-in-time procurement and usage for just as long as needed.

Self-Service

One of the hallmarks of the public cloud is the easy-to-use, remotely accessible interface based on modern web standards. All large public cloud service providers offer at least three interfaces: a web-based, graphical, point-and-click dashboard; a console-based command-line tool; and APIs. These enable customers to deploy and terminate IT resources anytime. These facilities make it easy for customers to perform self-service and further reduce time to market. In a traditional setting, even if IT deployment is outsourced to a third party, there is usually a lot of paperwork to be done, such as a request for quotes, purchase orders, and invoice processing.

Pay per Use

One of the promises of a public cloud is no lock-in through contracts. *No lock-in* means no upfront fees, no contractual time period, no early termination penalty, and no disconnection fees. Customers can move to another public cloud provider or simply take things onsite.

Google Cloud Platform adopts this definition and charges no upfront fees, has no contractual time period, and certainly charges no termination/disconnection fees. But Amazon Web Services offers a contract-like reservation plan that requires an initial payment to reserve resources and have lower usage costs during the reservation period. The downside of this reservation plan is that the promised savings are realized only if the same resource type is used nonstop the entire time.

The pay-per-use business model of a public cloud allows a user to pay the same for 1 machine running for 1,000 hours as they would for 1,000 machines running for 1 hour. Today, a user would likely wait 1,000 hours or abandon the project. In a public cloud, there is virtually no additional cost to choosing 1,000 machines and accelerating the user's processes.

WHAT IS SCALABILITY?

Scalability is a process through which an existing resource can be expanded on an on-demand basis either vertically or horizontally. An example of vertical scalability would be to upgrade a server's RAM from 2GB to 4GB, whereas horizontal scalability would add a second server with 2GB RAM. Scalability can be automatic or manual, but the end user should be able to update resources on an on-demand basis using either a web-based dashboard or an API.

Uncertain Growth Patterns

All organizations wish for exponential growth, but they can't commit sufficient IT infrastructure because they are not certain about the future. In a traditional setup, such scenarios result in unused capacity when growth is less than predicted or result in unhappy customers when the installed capacity is not able to handle additional load. Arbitrary loads are best handled by using public cloud deployments.

Why Google Cloud Platform?

Google Cloud Platform is built on the same world-class infrastructure that Google designed, assembled, and uses for corporate products like Google search, which delivers billions of search results in milliseconds. Google has also one of the largest, most geographically widespread, most advanced computer networks in the world. Google's backbone network comprises thousands of miles of fiber-optic cable, uses advanced software-defined networking, and is coupled with edge-caching services to deliver fast, consistent, scalable performance. Google is also one of the few companies to own a private fiber-optic cable under the Pacific Ocean.

Google Cloud Platform empowers software application developers to build, test, deploy, and monitor applications using Google's highly scalable and reliable infrastructure. In addition, it enables system administrators to focus on the software stack while allowing them to outsource the challenging work of hardware assembly, maintenance, and technology refreshes to experts at Google.

Hardware Innovations

Whereas a typical cloud service provider's strategy is wholesale-to-retail using standard hardware and software components, Google's approach has been to innovate at every level: hardware, networking, utilities, and software. This is evident from the multitude and variety of innovations that Google has introduced over the years. Needless to say, Google Cloud Platform benefits from all these innovations and thus differentiates itself from the competition:

- *Highly efficient servers:* In 2001, Google designed energy-efficient servers using two broad approaches: it removed unnecessary components like video cards, peripheral connections, and casing; and it used energy-efficient power supplies (that do AC-to-DC conversion) and power regulators (DC-to-DC conversion) and backup batteries on server racks.
- *Energy-efficient data centers:* In 2003, Google designed portable data centers using shipping containers that held both servers and cooling equipment. This modular approach produced better energy efficiency compared to traditional data centers at the time. Since 2006, Google has achieved the same efficiency using alternate construction methods.
- *Carbon neutrality:* In 2007, Google became a carbon-neutral Internet company, and it remains so today. Its data centers typically use 50% less energy compared to traditional data centers.
- *Industry-leading efficiency:* The cost of electricity is rapidly increasing and has become the largest element of TCO (currently 15%–20%). Power usage effectiveness (PUE) tends to be significantly lower in large facilities than in smaller ones. Google's data centers have very low PUE: it was 1.23 (23% overhead) in Q3 2008 and came down to 1.12 (12% overhead) in Q4 2014. This is significantly lower than the industry average of 1.7 (70% overhead).

All of these hardware innovations result in lower operational costs for Google, and the difference is passed to Google Cloud Platform users. This means customers save on costs.

Software Innovations

Infrastructure innovation is not just about hardware. Google has also led the industry with innovations in software infrastructure:

- *Google File System:* In 2002, Google created the Google File System (GFS), a proprietary distributed file system designed to provide efficient, reliable access to data using a large cluster of commodity hardware.
- *MapReduce:* In 2004, Google shared the MapReduce programming model that simplifies data processing on large clusters. The Apache Hadoop project is an open source implementation of the MapReduce algorithm that was subsequently created by the community.

- *BigTable:* In 2006, Google introduced the BigTable distributed storage system for structured data. BigTable scales across thousands of commodity servers and is used by several Google applications.
- *Dremel:* In 2008, Google shared the details of a system called Dremel that has been in production since 2006. Dremel is a scalable, interactive, ad hoc query system for analyzing read-only nested data that is petabytes in size. Dremel combines multilevel execution trees, uses a columnar data layout, and is capable of running aggregation queries over trillion-row tables in seconds. Dremel is the backend of Google BigQuery.
- *Pregel:* In 2009, Google created a system for large-scale graph processing. The principles of the system are useful for processing large-scale graphs on a cluster of commodity hardware. Examples include web graphs, among other things.
- *FlumeJava:* In 2010, Google introduced FlumeJava. FlumeJava is a pure Java library that provides a few simple abstractions for programming data-parallel computations. These abstractions are higher-level than those provided by MapReduce and provide better support for pipelines. FlumeJava makes it easy to develop, test, and run efficient data-parallel pipelines of MapReduce computations.
- *Colossus:* In 2010, Google created the successor to GFS. Details about Colossus are slim, except that it provides a significant performance improvement over GFS. New products like Spanner use Colossus.
- *Megastore:* In 2011, Google shared the details of Megastore, a storage system developed to meet the requirements of today's interactive online services. Megastore blends the scalability of a NoSQL datastore with the convenience of a traditional RDBMS in a novel way, and provides both strong consistency guarantees and high availability. Megastore provides fully serializable ACID semantics within fine-grained data partitions. This partitioning allows Megastore to synchronously replicate each write across a wide area network with reasonable latency and support seamless failover between datacenters.
- *Spanner:* In 2012, Google announced this distributed database technology. Spanner is designed to seamlessly operate across hundreds of datacenters, millions of machines, and trillions of rows of information.
- *Omega:* In 2013, Google introduced Omega—a flexible, scalable scheduler for large-scale compute clusters. Google wanted to move away from current schedulers, which are monolithic by design and limit new features. Omega increases efficiency and utilization of Google's compute clusters.
- *Millwheel:* In 2014, Google introduced Millwheel, a framework for fault-tolerant stream processing at Internet scale. Millwheel is used as a platform to build low-latency data-processing applications within Google.

All of these innovations are used to make Google Cloud Platform products, just as they are used to build Google's internal products. By using Google Cloud Platform, customers get faster access to Google innovations, thereby distinguishing the effectiveness of applications hosted on Google Cloud Platform.

Figure 1-1 shows a few important innovations from the above list to help visualize the continuous innovations by Google.

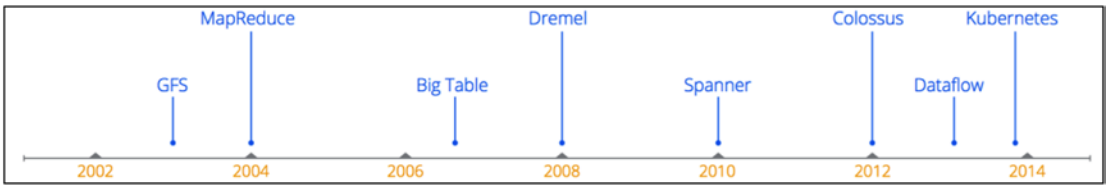

Figure 1-1. *Google's software innovations that are actively used in Google Cloud Platform*

Economic Innovations

In addition to making technical and infrastructure innovations, Google has also taken a fresh look at how to charge for cloud computing resources. Let's consider the economic innovations that Google has introduced in Google Cloud Platform, many of which benefit cloud platform users.

Typical public cloud providers, in particular Amazon Web Services, provide two types of pricing options for products: *on-demand* and *reserved* pricing. The guiding principle behind these two types of pricing options is to secure longer-term commitments from users. In the on-demand pricing model, the customer is free to use the resource for as long as needed and is free to leave anytime. There is no time contract or penalty for termination; this is typical of cloud hosting. In the reserved price model, the customer is required to pay a nonrefundable upfront fee and select the type of resource. As a result, the customer enjoys lower hosting charges for the specified time period.

There are several shortcomings in the reserved pricing model. First, because lower pricing is tied to the resource type, if the customer decides to switch resource types (say, due to different traffic patterns than expected), they are thrown back to the higher pricing model. Second, the upfront fees are time bound and not based on the number of hours of usage. Third, the upfront fees are not refundable if the customer decides to terminate early. In essence, the onus of choosing the right resource type and time duration is with the customer; there is no reconciliation if the actual workload is different from the expected workload.

Google's approach is that customers should be want to host on Google Cloud Platform due to its meritocracy and technical superiority. They should be able to leave anytime and not be tied through contract-like approaches. They should also be able to switch resource types anytime, as their needs change. Finally, while customers are hosting on Google Cloud Platform, they should enjoy the best pricing, on par with the industry.

To realize these objectives, Google has created a new type of pricing model called a *sustained-use discount*. Under this model, Google Cloud Platform automatically applies discounts to resources that run for a significant time. The discount is based on the cumulative amount of time a resource of a particular type is up rather than being tied to a single instance. This means two instances of equivalent specs running simultaneously or concurrently are given the same discount as long as the cumulative hosting period is above a threshold. Sustained-use discounts combined with per-minute billing ensure that customers get the best deal. The following list shows the sustained-use discounts as of this writing (March 2015):

- *0%–25%:* 100% of base rate
- *25%–50%:* 80% of base rate
- *50%–75%:* 60% of base rate
- *75%–100%:* 40% of base rate

Google has ventured to decipher the sometimes-complex world of cloud pricing by explaining how to calculate the cost of a cloud deployment. See the following post in the official Google cloud platform blog for details: `http://googlecloudplatform.blogspot.sg/2015/01/understanding-cloud-pricing.html`.

A Quick Comparison to AWS

This section highlights a few select features of Google Cloud Platform and how they compare with the incumbent public cloud provider, Amazon Web Services:

- Google Compute Engine, the Internet-as-a-service (IaaS) product from Google Cloud Platform, adopts a per-minute charging model except for the initial minimum 10-minute tier. On the other hand, AWS charges on an hourly-basis.

  Let's consider two example use cases. First, if you use an instance for 11 minutes, you pay for 11 minutes in Google Cloud Platform, but you pay for 60 minutes with Amazon Web Services. Second, if you use an instance for 1 minute, you pay for 10 minutes in Google Cloud Platform or 60 minutes in Amazon Web Services. In either case, you can see that Google Cloud Platform is cheaper than Amazon Web Services.

- Google Compute Engine is better suited to handle traffic spikes. This is because the Compute Engine load balancers don't require pre-warming, unlike AWS load balancers. In addition, pre-warming AWS load balancer requires customers to subscribe to AWS support. Compute Engine load balancers are able to scale instantly when they notice a sudden traffic spike.

  In 2013, Google demonstrated that its load balancers could serve 1 million requests per second on a sustained basis and within 5 seconds after setup. You are advised to read the full article at http://googlecloudplatform.blogspot.in/2013/11/compute-engine-load-balancing-hits-1-million-requests-per-second.html.

- Compute Engine's persistent disks (PDs) support a larger disk size (currently 10TB) compared with AWS. In addition, Google includes the I/O costs in the cost of the PD, thereby giving customers predictable costing. In the case of AWS, the cost of I/O is separate from the cost of the raw disk space. Moreover, other nice features include the ability to mount a PD to multiple VMs as read-only or a single VM in read-write mode.

- Compute instances are hosted as virtual machines in IaaS. Periodically, the IaaS service provider needs to do maintenance (host OS or hardware) on the platform. The hardware may also fail occasionally. In such cases, it is desirable to have the VM automatically migrate to another physical host. Compute Engine can do live migration.

- Google App Engine, the platform-as-a-service (PaaS) product from Google Cloud Platform, is in our view a pure PaaS product when compared with Beanstalk from Amazon Web Services. This is because Beanstalk is a management layer built on top of AWS EC2. The implication of this design choice is that Beanstalk needs to have at least one EC2 instance up all the time, which adds to hosting costs. App Engine, on the other hand, charges only when there is traffic and includes a monthly free tier.

- BigQuery, the big-data analytics product from Google Cloud Platform, is an integrated and fully hosted platform that scales to thousands of nodes and charges only for space and computation time. In comparison, the AWS equivalent (Red Shift) requires users to configure the system and also charges by the hour rather than based on usage.

- Google data centers (that host Google Cloud Platform's regions and zones) are spread globally and interconnected by Google's private fiber network. This means network traffic between two cloud platform zones passes through a private network and not over the public Internet. As of today, AWS does not have such a private network.

Overall, Google's approach with Google Cloud Platform is not to achieve feature parity with Amazon Web Services but to build products that are by far the best in the industry and in the process fill in the gaps in the AWS portfolio. Hence, the question to ask is whether your needs are being met by what Google Cloud Platform has today, rather than talking about what Google Cloud Platform doesn't have.

When talking about the strengths of Google Cloud Platform, it is important to acknowledge that Amazon Web Services currently has a broader portfolio of products and services than Google Cloud Platform. This is primarily due to the fact that AWS started much earlier, while Google was busy building the fundamentals right, as shown in the list of major software innovations earlier in this chapter.

Summary

We started this chapter by defining the concept of cloud computing. Following this, we leaped into public clouds, which we cover in this book. We shared with you the advantages of a public cloud from several perspectives: technical, economic, and business. Following this, we highlighted several Google research publications that are used to build the strong foundation of Google Cloud Platform. We concluded this chapter by listing the strengths of Google Cloud Platform when compared with Amazon Web Services.

The promise of the public cloud is not just cheaper computing infrastructure, but also faster, easier, more flexible, and ultimately more effective IT.

CHAPTER 2

Getting Started with Google Cloud Platform

Welcome to Google Cloud Platform!

Cloud Platform is a set of modular cloud-based services that provide building blocks you can use to develop everything from simple web sites to sophisticated multitier web-based applications. This chapter introduces the core components of Cloud Platform and guides you through the process of getting started with it.

Cloud Platform Building Blocks

This section gives you an overview of the products in Cloud Platform and explains the technology clusters they belong to. This approach will help you select which chapters of this book you need to read to quickly get started with Cloud Platform. We do, however, encourage you to read the book cover to cover!

Projects

Projects are top-level containers in Cloud Platform. Using projects, you can consolidate all related resources, IT and non-IT, on a project-by-project basis. This enables you to work on several projects at the same time while ensuring that the resources are in separate control domains. Each project is identified by a tuple consisting of the following three items:

- *Project name*: This is a text field that lets you store a friendly, descriptive string about the project's purpose. This is only for your reference and can be changed any number of times during the project's lifetime.
- *Project ID*: The Project ID is a globally unique string across all Cloud Platform products. A random project ID, made of three words delimited by hyphens between them, will be automatically generated during project creation. You can change the suggested ID as long as it's unique across all Cloud Platform projects from all Cloud Platform users. Project ID can include lowercase letters, digits, or hyphens, and it must start with a lowercase letter. Once the choice is made, the ID cannot be changed during the project's lifetime.
- *Project number*: Cloud Platform automatically assigns a project number at creation time for the project's lifetime. You have no control over this number.

The command-line developer tool called `gcloud` (described later) requires a project ID for identifying and accessing various IT resources. Public-facing Cloud Platform APIs may require either the project ID or the project number for resource-identification purposes. Cloud Platform uses project numbers almost exclusively to identify projects.

In addition to IT resources, a Cloud Platform project also stores information about billing and authorized users. In Cloud Platform, a billing account is considered separate from a project account. One billing can be linked to more than one project account. A billing account is identified by a set of the following four items:

- *Billing account ID*: This is automatically generated by Google billing. You don't have any control over it and don't need to worry about it.
- *Billing account name*: Tis a friendlier description of the billing account. You can set it during account creation and change it any time during the account's lifetime.
- *Status*: The status of a billing account is either active or closed.
- *# Of projects*: Each billing account, after being created, is attached to projects. One billing account can be attached to one or more projects, whereas one project can be attached to only one billing account.

By using projects, you can provide services to different customers and separate the associated costs. Cloud Platform generates a separate bill for each project. At the same time, you can pay for all your projects using the same billing account.

As of this writing, a project can only be created using the web-based Developers Console, not with the `gcloud` command-line tool or the Cloud Platform API. You also can't list all the projects associated with a Google account using `gcloud` or an API. This restriction is in place because the project-creation feature is not part of the public-facing APIs, which are also used by `gcloud`. However, you can store project information using `gcloud` and use it automatically for subsequent requests. You can create a project by visiting `http://console.developers.google.com` and filling in the required details.

Regions, Zones, Resources, and Quotas

Cloud Platform resources are hosted in multiple locations worldwide. These locations are composed of *regions*, and each region is further broken into *zones*. A zone is an isolated location within a region. Zones have high-bandwidth, low-latency network connections to other zones in the same region.

Cloud Platform resources can be classified as *global*, *regional*, or *zonal*. IT resources in the same region or zone can only use resources that are specific to the region or zone. For example, Compute Engine, the Infrastructure-as-a-Service product from Cloud Platform, instances and persistent disks are both zonal resources. If you want to attach a persistent disk to an instance, both resources must reside in the same zone. Similarly, if you want to assign a static IP address to a Compute Engine instance, the instance must reside in the same region as the static IP. Not all resources are region or zone specific; some, such as disk images, are global resources that can be used by any other resources at any location.

During the resource-creation stage, depending on the scope of the resource, Cloud Platform prompts you to choose either a region or a zone. For example, when you create an instance or disk, you are prompted to select a zone where that resource should serve traffic. Other resources, such as static IPs, live in regions; when you select a region, the system chooses an appropriate regional IP address.

Cloud Platform makes it easy to programmatically query for current regions and zones and to list all of a region's or zone's public details. Although regions and zones do not change frequently, Google wants to make it easy for you to retrieve this information without having to browse through a web site or documentation. Let's look at how to use the `gcloud` command-line tool to query information about regions and zones. For now, focus on the results; you learn about `gcloud` later.

All generally available Cloud Platform resources that have regional scope, such as Compute Engine, are available in all regions/zones. For products that have global scope, such as App Engine and BigQuery, you do not need to select a region or zone. Let's list the regions where Compute Engine (and, by extension, persistent disks, load balancers, autoscalers, Cloud Storage, Cloud Datastore, and Cloud SQL) is available, using `gcloud`:

```
$ gcloud compute regions list
NAME            CPUS          DISKS_GB     ADDRESSES RESERVED_ADDRESSES STATUS TURNDOWN_DATE
asia-east1      2.00/24.00    10/10240     1/23      1/7                UP
europe-west1    0.00/24.00     0/10240     0/23      0/7                UP
us-central1     0.00/24.00     0/10240     0/23      0/7                UP
```

This output shows that there are currently three regions in Cloud Platform, one on each major continent. This choice was made strategically to accommodate applications and data that need to reside on the respective continent.

In addition to the regions, the previous output shows quota information. A *quota* in Cloud Platform is defined as a soft limit for a given type of resource. If you need more than the stated limit, you can request additional resources by filling out an online Google form. The previous output shows that this particular Google account has instantiated two CPUs, has a 10BG persistent disk, and is using two public IPs, one of which is a reserved IP address. All regions are operating normally, and there is no announced teardown date for any of them.

Let's examine one of the regions in detail:

```
$ gcloud compute regions describe asia-east1
creationTimestamp: '2014-11-18T14:51:15.377-08:00'
description: asia-east1
id: '1220'
kind: compute#region
name: asia-east1
quotas:
- limit: 24.0
  metric: CPUS
  usage: 2.0
- limit: 10240.0
  metric: DISKS_TOTAL_GB
  usage: 10.0
- limit: 7.0
  metric: STATIC_ADDRESSES
  usage: 1.0
- limit: 23.0
  metric: IN_USE_ADDRESSES
  usage: 1.0
- limit: 1024.0
  metric: SSD_TOTAL_GB
  usage: 0.0
- limit: 1500.0
  metric: LOCAL_SSD_TOTAL_GB
  usage: 0.0
- limit: 240.0
  metric: INSTANCES
  usage: 0.0
```

```
selfLink: https://www.googleapis.com/compute/v1/projects/www-redcross-sg/regions/asia-east1
status: UP
zones:
- https://www.googleapis.com/compute/v1/projects/www-redcross-sg/zones/asia-east1-a
- https://www.googleapis.com/compute/v1/projects/www-redcross-sg/zones/asia-east1-b
- https://www.googleapis.com/compute/v1/projects/www-redcross-sg/zones/asia-east1-c
```

This output shows more interesting and useful information about the region. First, you can see that Google publicly discloses when this zone went live (or was upgraded). Second, just like any other entity in Cloud Platform, the region has an ID, a name, and a description. Finally, the output states that the region contains three zones.

Let's now list all the zones in all the regions in Cloud Platform:

```
$ gcloud compute zones list
NAME           REGION       STATUS NEXT_MAINTENANCE TURNDOWN_DATE
asia-east1-a   asia-east1   UP
asia-east1-c   asia-east1   UP
asia-east1-b   asia-east1   UP
europe-west1-b europe-west1 UP
europe-west1-c europe-west1 UP
europe-west1-d europe-west1 UP
us-central1-f  us-central1  UP
us-central1-a  us-central1  UP
us-central1-c  us-central1  UP
us-central1-b  us-central1  UP
```

This output shows that there are a total of 10 zones across 3 regions. Of course, this is as of this writing; Google is expected to add new regions and zones regularly.

From the region and zone names, you can decipher that the fully qualified name for a zone is made up of `<region>-<zone>`. For example, the fully qualified name for zone a in region `us-central1` is `us-central1-a`.

Let's look at the details for one particular zone:

```
$ gcloud compute zones describe asia-east1-a
creationTimestamp: '2014-05-30T18:35:16.575-07:00'
description: asia-east1-a
id: '2220'
kind: compute#zone
name: asia-east1-a
region: https://www.googleapis.com/compute/v1/projects/www-redcross-sg/regions/asia-east1
selfLink: https://www.googleapis.com/compute/v1/projects/www-redcross-sg/zones/asia-east1-a
status: UP
```

Just like a region, a zone has a creation date, an ID, a kind, and a name.

The Developers Console

The Developers Console is a web-based interface that you can use to create and manage your Cloud Platform resources. You can also view and manage projects, team members, traffic data, authentication, and billing through the Developers Console; see `https://developers.google.com/console/help/new` to learn about its capabilities. Figure 2-1 shows the Google Developers Console overview screen.

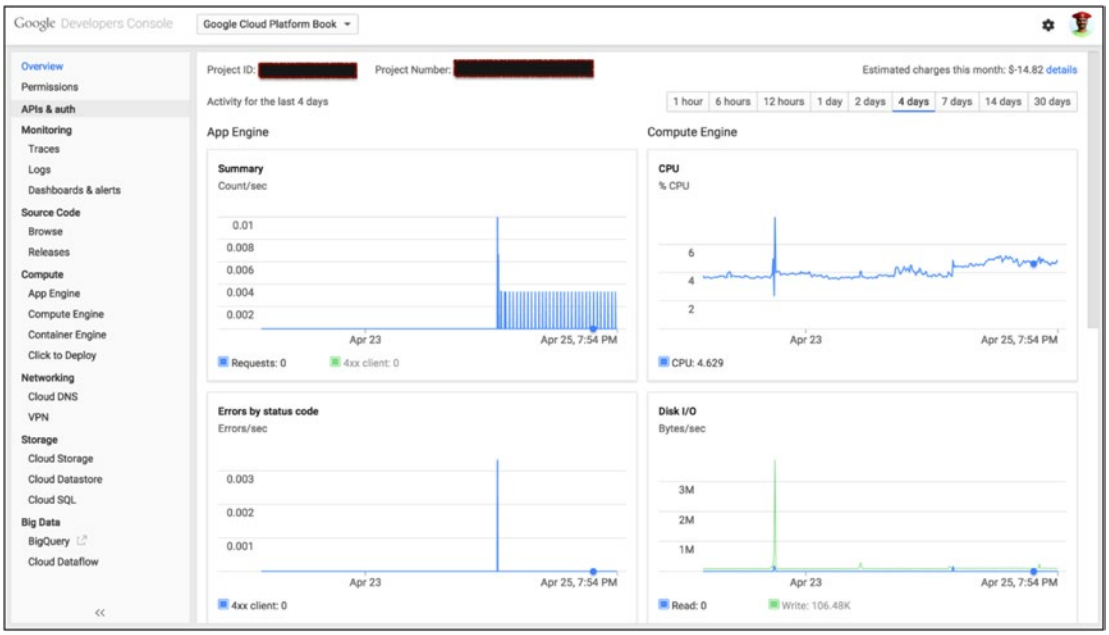

Figure 2-1. *Google Developers Console*

This section looks at some of the Developers Console functionality that is generally applicable for deploying Cloud Platform products.

Permissions and Auth

Each Cloud Platform project can be accessed by one or more Google accounts. The Google account that creates a project is automatically designated as its owner. In addition to an owner, two other roles are allowed that have different levels of access to a project:

- *Owner*: An owner can change project settings and manage team members.
- *Editor*: An editor can change project settings.
- *Viewer*: A viewer can read all project settings and information.

The owner, using the web-based Developers Console, can add additional owners, editors, and viewers. To do so, choose Developers Console ➤ Permissions ➤ Add Member, as shown in Figure 2-2. In addition to regular Google accounts (which are accessed by humans), Cloud Platform also supports a category called Service Accounts. These are automatically added by Cloud Platform and are used to authenticate the project to other Google services and APIs.

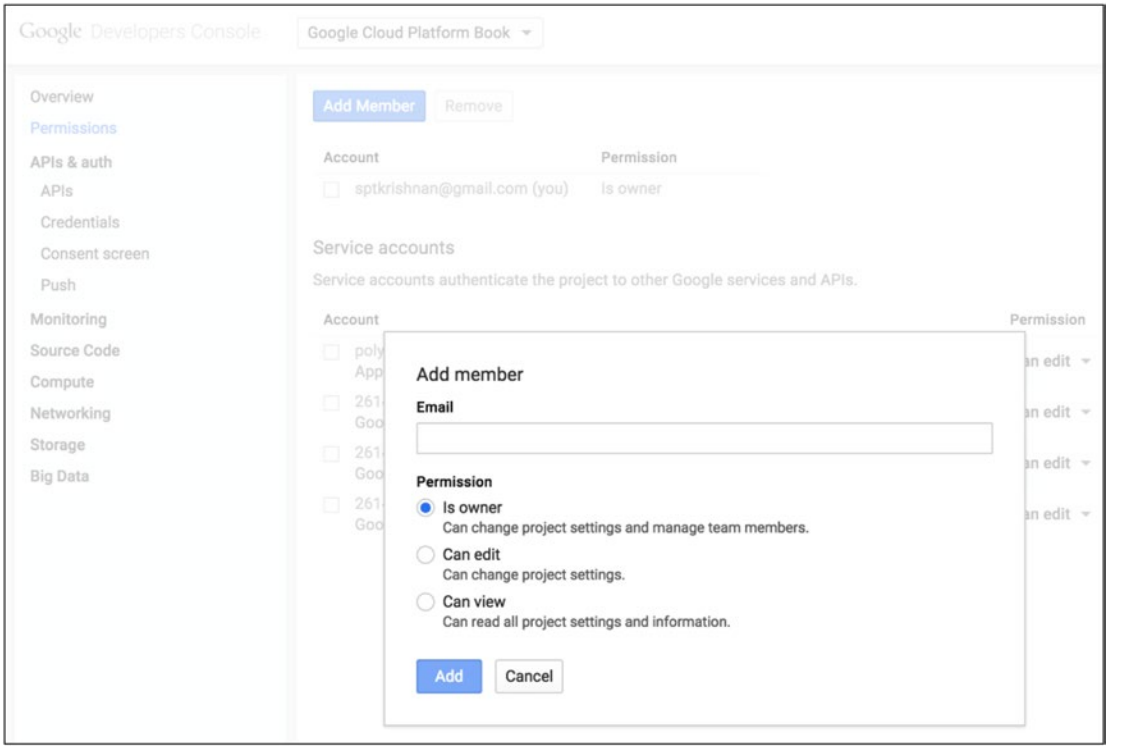

Figure 2-2. *Adding team members to a project*

Permissions allow a project's resources to access various Cloud Platform APIs. Some APIs allow unlimited and unmetered access, such as the Compute Engine API. Other APIs impose daily quotas and access-rate limits. Auth (short for authentication) allows one or more client applications to access APIs that have been enabled in a particular project. In addition, it lets applications access your private data (for example, contact lists). We examine the OAUTH technology in Chapter 3. For now, you just need to know how to create new client ID or key using the Developers Console. Go to Developers Console ➤ APIs & Auth ➤ Credentials to create an OATH2 client ID or a public API access key, as shown in Figure 2-3.

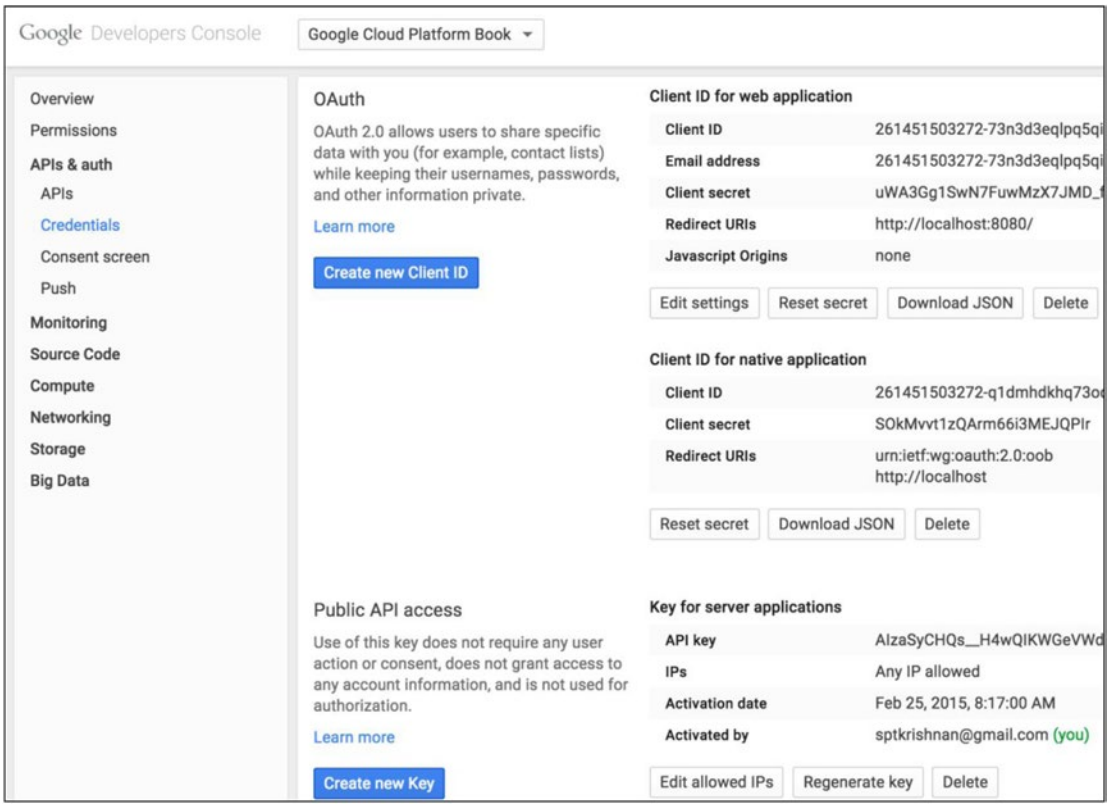

Figure 2-3. *Creating new credentials*

When you use the version of OAUTH called *three-legged authentication (3LO)*, your users are shown a consent screen that they need to accept before Google will authorize your application to access their private data. This is explained in the OAUTH section in Chapter 3. For now, to customize the consent screen in the Developers Console, choose Developers Console ➤ APIs & Auth ➤ Consent Screen as shown Figure 2-4.

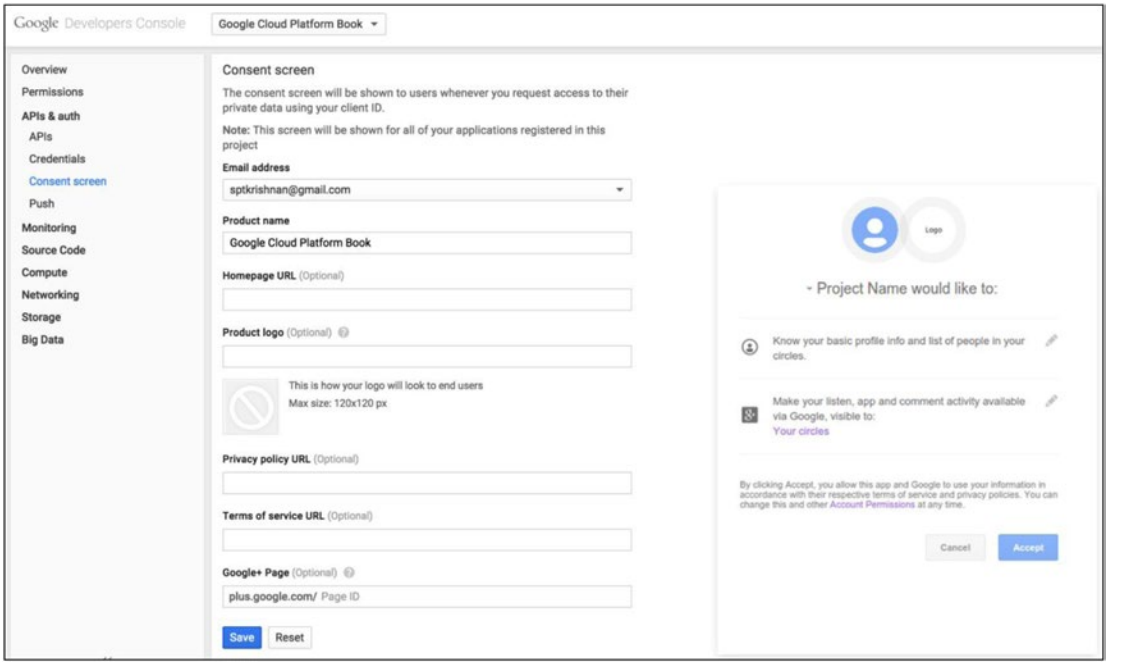

Figure 2-4. *Consent screen setup and customization*

The Cloud SDK and the gcloud Tool

The Google Cloud SDK contains tools and libraries that enable you to easily create and manage resources on Cloud Platform. It runs on Windows, Mac OS X, and Linux, and it requires Python 2.7.x or greater or another language runtime for language-specific support in the SDK. Installing the Cloud SDK is operating system dependent and is well documented at https://cloud.google.com/sdk. Follow the instructions there to install the Cloud SDK.

The most common way to manage Cloud Platform resources is to use the `gcloud` command-line tool. `gcloud` is included as part of the Cloud SDK. After you have installed the Cloud SDK, you need to authenticate the `gcloud` tool to access your account. Run the command `gcloud auth login` to do this, as follows:

```
$ gcloud auth login
Your browser has been opened to visit:
https://accounts.google.com/o/oauth2/auth?redirect_uri=http%3A%2F%2Flocalhost%3A8085%2F&
prompt=select_account&response_type=code&client_id=32555940559.apps.googleusercontent.com&
scope=https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fuserinfo.email+https%3A%2F%2F
www.googleapis.com%2Fauth%2Fcloud-platform+https%3A%2F%2Fwww.googleapis.com%2Fauth%2F
appengine.admin+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcompute&access_type=offline

Saved Application Default Credentials.

You are now logged in as [cloudplatformbook@gmail.com].
Your current project is [cloud-platform-book].  You can change this setting by running:
  $ gcloud config set project PROJECT
```

`gcloud` opens a new browser window when you execute this command. After you click Accept, control returns to the `gcloud` tool, and your `gcloud` instance is configured to access your Google account and project. If you would like to switch to another account or project, you can use the following commands (replacing the account and project values):

```
$ gcloud config set account cloudplatformbook@gmail.com
$ gcloud config set project cloud-platform-book
```

`gcloud` has a comprehensive built-in help system. You can request help at multiple levels. Here are a few examples:

- `gcloud -h`: Produces help at the outermost level. The tool lists various command groups, commands, and optional flags that are permissible.
- `gcloud compute -h`: Lists the command groups, commands, and optional flags that apply to Google Compute Engine.
- `gcloud compute instances -h`: Lists the commands and optional flags that apply to the `instances` command group in Google Compute Engine.

This way, you can request help at multiple levels. To learn about all of `gcloud`'s features, visit https://cloud.google.com/sdk/gcloud. You can list the various components supported in gcloud by using the command `gcloud components list`.

APIs and Cloud Client Libraries

Google follows an API-first development philosophy, and APIs are the primary developer interface for Google's products, including Cloud Platform. Hence, before you can use a product—say, Compute Engine—you need to enable that particular API in your project. API enablement is on a project-by-project basis. Google makes it easy for you to enable a particular API using the Developers Console. You can access the APIs section by choosing Developers Console ➤ APIs & Auth ➤ APIs. The tabbed screen shows the list of all available APIs and the APIs that have been enabled in a project. Figure 2-5 shows a subset of the APIs available, and Figure 2-6 shows the APIs that have been enabled for this project.

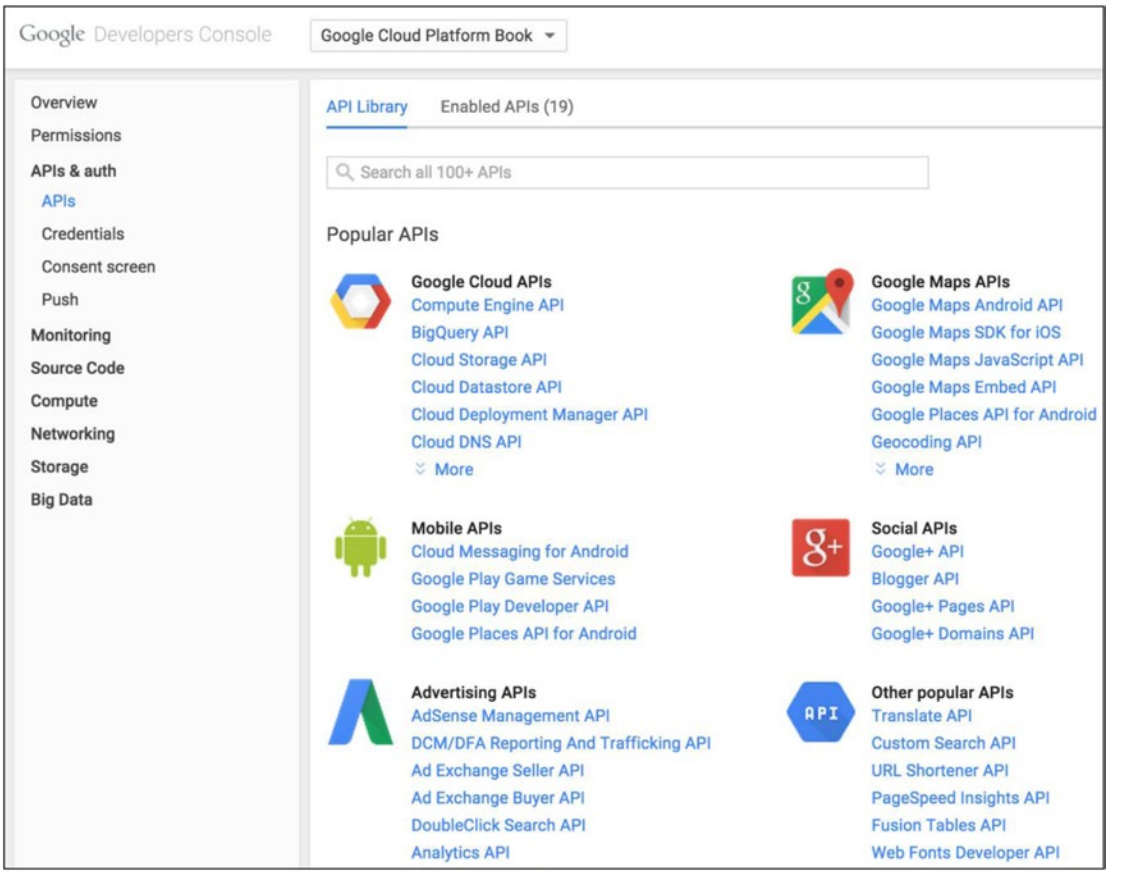

Figure 2-5. *Subset of APIs available to Google developers*

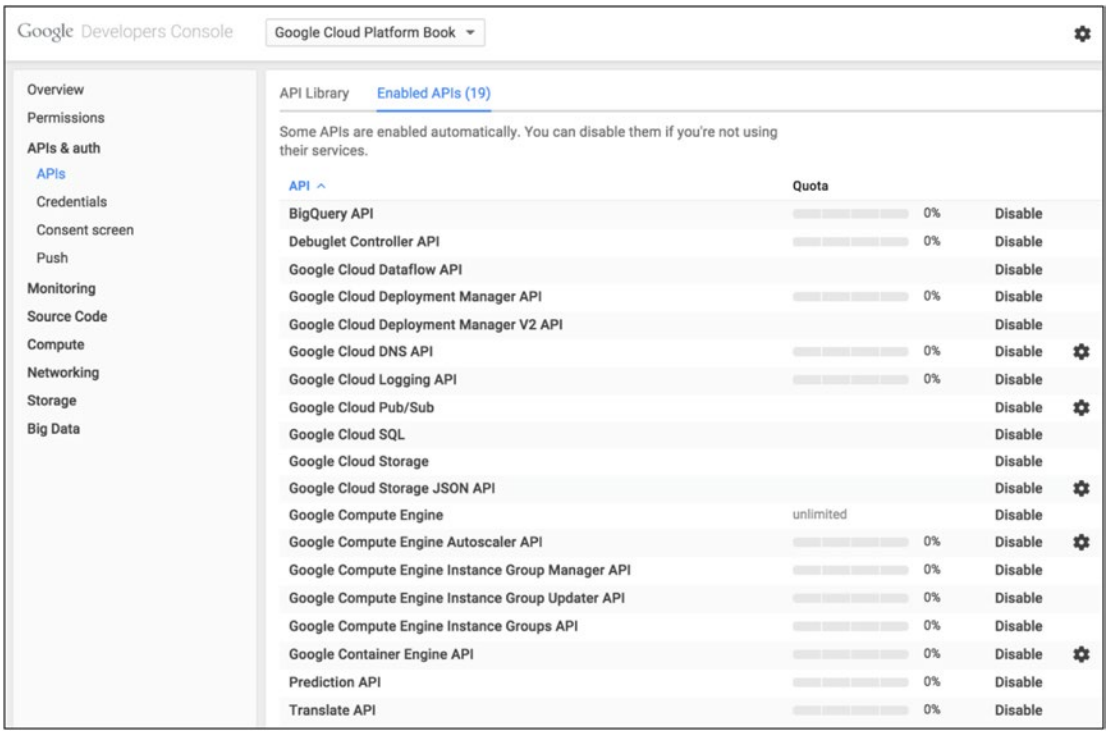

Figure 2-6. *List of APIs enabled in one project*

Deploying resources on demand and releasing them when they aren't needed realizes the power of the Cloud Platform. This workflow can be achieved using several methods. When you use the Developers Console, the response time is slow and the process is manual. When you use the `gcloud` tool, the response time is faster, and you can automate the process by using a script. However, Google designed `gcloud` to be used by developers and not programs, so you have to write code to parse the command output. You can use the Cloud Platform APIs to allocate and release resources as needed, but because the APIs are RESTful and stateless, you need to maintain state between API calls.

Cloud Client libraries fill the gap of programmatically accessing the Cloud Platform while integrating into the respective programming language so that the client can use other language features. The Cloud Platform APIs have been implemented as library functions in several programming languages. As of this writing, Google officially supports the Python, Node.js, and Go languages.

Cloud Platform Products

This section describes the various Cloud Platform technologies covered in this book. We hope this overview will guide you on your journey into Cloud Platform:

- Compute
  - *Compute Engine*: Compute Engine is an infrastructure as-a-service (IaaS) product. Using it, you can launch virtual machines, create networks, and attach local and remote persistent disks based on magnetic or solid state technologies. You can also design and build advanced architectures that include load-balancing and auto-scaling and that span multiple zones in a region or multiple geographical regions worldwide. Compute Engine gives you maximum flexibility and is primarily targeted at architects and system administrators.
  - *App Engine*: App Engine is a platform as a service (PaaS) product. Using it, you can build web-scale, autoscaling applications. App Engine is targeted at software developers and provides a comprehensive collection of libraries. Using it, you can simply upload an application to the platform, and App Engine takes care of everything else.
  - *Container Engine*: Containerized applications are being explored as the next step in DevOps standard operating procedures and the next generation of application development. Docker is at the forefront of this revolution and is building an industry-wide consensus about the format and interface of application containers. An application container is enabled by a set of core innovations in the Linux kernel that Google invented almost a decade ago. This places Google at the forefront of driving container adoption among developers. Container Engine is covered in Chapter 6; it is still in an early stage of evolution.
  - *Managed VMs*: Managed virtual machines are the next generation of App Engine and feature many new capabilities such as Docker-formatted application containers, writable local disks, and live debugging of applications over SSH. Whereas Container Engine enables you to build sophisticated multi-tier applications where each node is a Docker container, managed VMs take care of all of them. In essence, Container Engine is an unmanaged platform for Docker-based applications, and a managed VM is a managed platform for Docker-based applications. Managed VMs are also covered in Chapter 6.
- Storage
  - *Cloud SQL*: Cloud SQL is a managed RDBMS product and is 100% binary compatible with open source MySQL server software. Google manages all the database-management tasks, and you can focus on building an app that needs a SQL back end. Cloud SQL supports advanced configurations such as read replicas (internal and external) and SSL connections.
  - *Cloud storage*: Cloud storage is object-based file storage that you can use to store data files without worrying about file system setup and maintenance. Cloud storage also includes automatic transparent global edge caching so that you don't have to set up another entity manually. Cloud storage offers different product flavors based on durability characteristics.
  - *Cloud Datastore*: Cloud Datastore is a managed, NoSQL, schemaless database for storing non-relational data. You can use this service to store key:value-based data. Cloud Datastore scales as your data needs increase, and you pay only for space that you consume.

- Big Data
  - *BigQuery*: BigQuery is a hosted Big Data analytics platform. BigQuery lets you query datasets that are multiple terabytes in size and features data ingestion at the rate of 100,000 rows per second per table.
  - *Cloud Pub/Sub*: Cloud Pub/Sub is a hosted messaging and queuing product that lets you connect multiple producers and consumers and enable low-latency, high-frequency data transfer between them.
  - *Cloud Dataflow*: Cloud Dataflow is a simple, flexible, powerful system you can use to perform data-processing tasks of any size. It lets you build, deploy, and run complex data-processing pipelines.
- Services
  - *Cloud Endpoints*: Cloud Endpoints enables you to create RESTful services and make them accessible to iOS, Android, and JavaScript clients. It also automatically generates client libraries to make wiring up the front end easy. With built-in features include denial-of-service protection, OAuth 2.0 support, and client key management, Cloud Endpoints lets you host API endpoints in Cloud Platform.
  - *Google APIs*: Applications can consume both Cloud Platform product APIs (for example Google Storage) and Google products APIs (for example Google Maps). This book includes an example of using the *Translate API* to translate content among 90 pairs of human languages.
- Networking
  - *Cloud DNS*: Cloud DNS is a reliable, resilient, low-latency DNS service from Google's worldwide network of Anycast DNS servers. You can manage your DNS records using the Developers Console UI, the `gcloud` command-line tool, or a full-featured RESTful API.
  - *Authentication*: Authentication is an essential step for governing access to your Cloud Platform resources or Google user data. Google uses the OAUTH 2.0 protocol exclusively for both authentication and authorization. We cover OAuth 2.0 and the various operational models in this book.
  - *Developer Toolbox*: Cloud Platform provides several tools to assist you in building, deploying, and maintaining awesome applications. We cover a few of them in this book, such as cloud repositories, container registries, click-to-deploy, and so on.

Summary

This chapter introduced you to the Cloud Platform's intricacies. We started by explaining the core building blocks of Cloud Platform, the various components of a project, and the steps you need to follow to get started.

We also explained the developer tools and gave a brief overview of the Cloud Platform products discussed in this book. Welcome aboard—let's get going!