Getting Started with Google Cloud Platform

2 authors, including:

Krishnan S. P. T.
REAN Cloud
23 PUBLICATIONS   **199** CITATIONS

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.
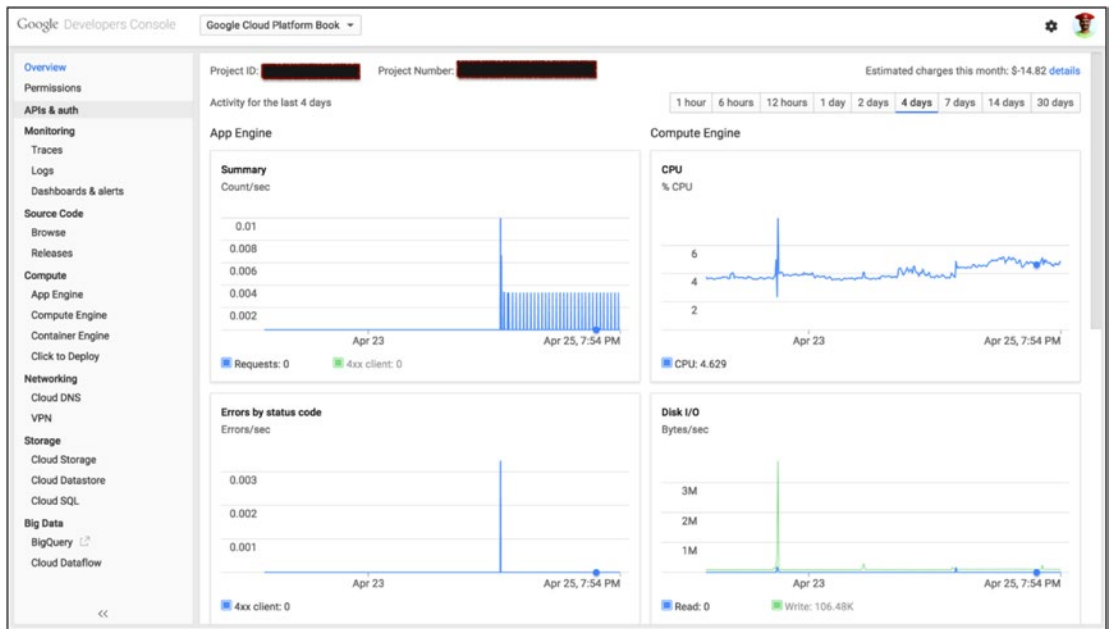
**Figure 2-1.** *Google Developers Console*

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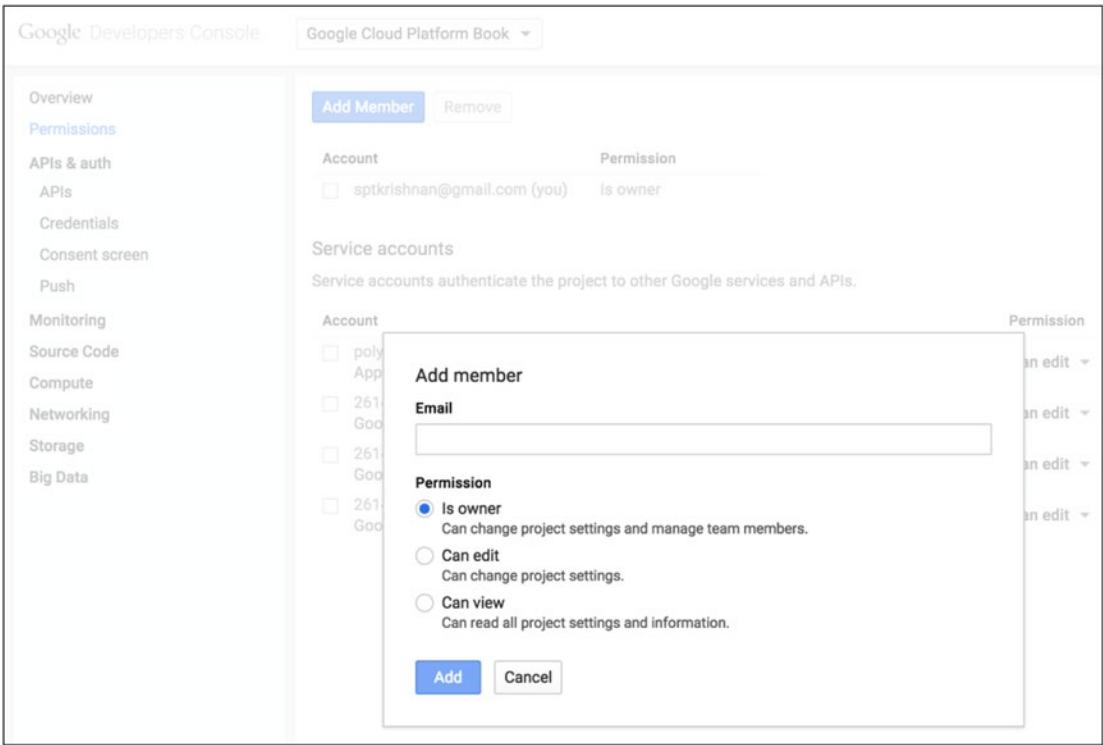
Figure 2-2. *Adding team members to a project*

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.
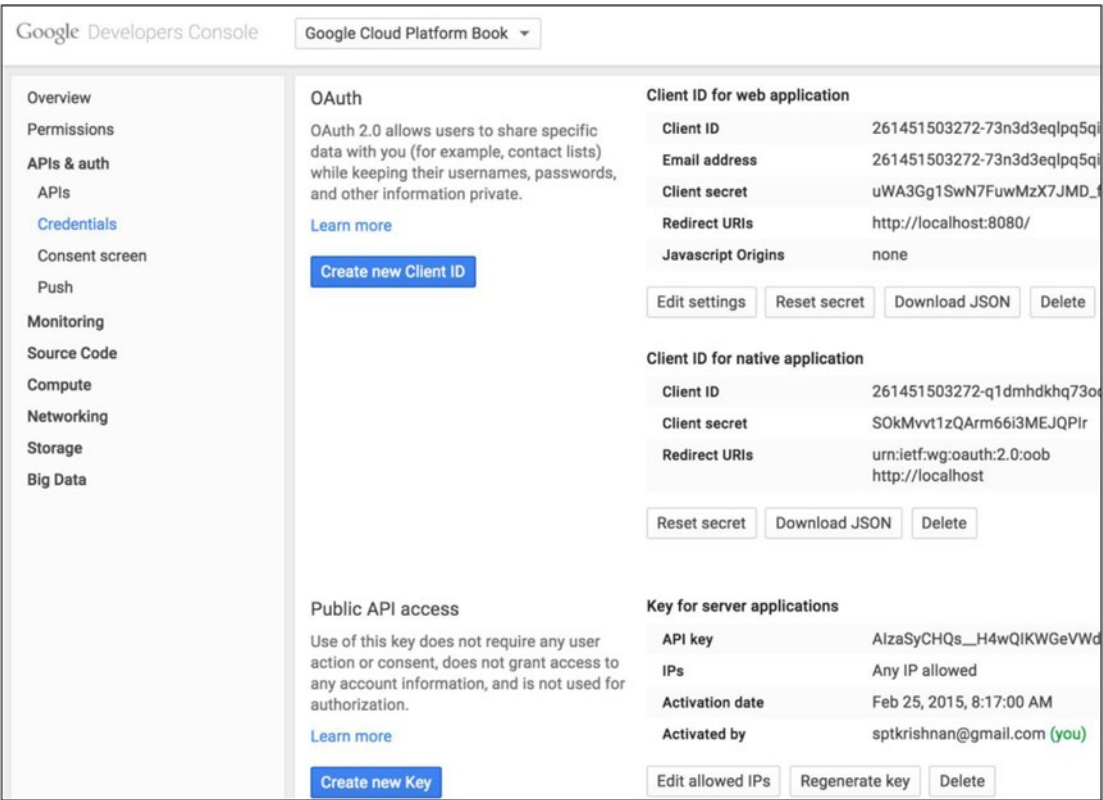
Figure 2-3. *Creating new credentials*

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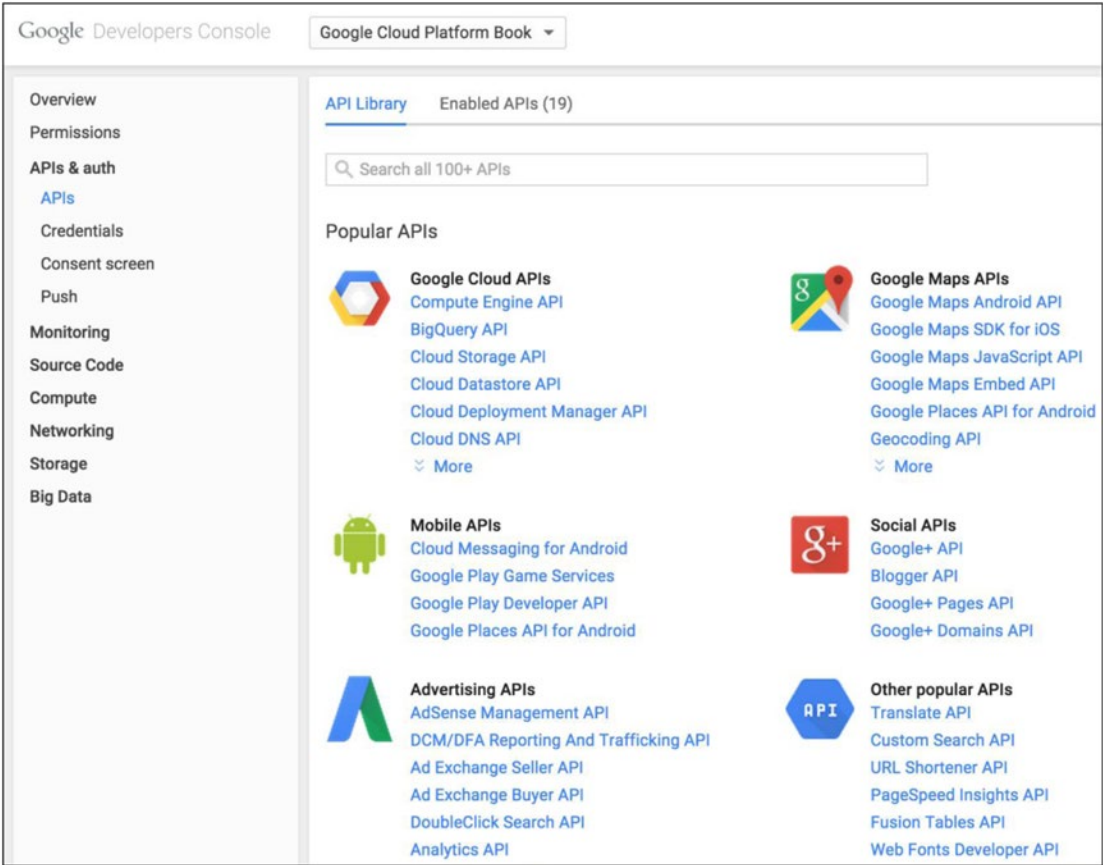
gcloud has a comprehensive built-in help system. You can request help at multiple levels. Here are a few examples:

- 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!