Google Cloud Platform All-In-One Guide

Get Familiar with a Portfolio of Cloud-based Services in GCP

Praveen Kukreti

bpb

Google Cloud Platform All-In-One Guide

Get Familiar with a Portfolio of Cloud-based Services in GCP

Praveen Kukreti

bpb

Google Cloud Platform All-In-One Guide

Get Familiar with a Portfolio of Cloud-based Services in GCP

Praveen Kukreti

www.bpbonline.com

First published: 2023

Published by BPB Online

WeWork

119 Marylebone Road

London NW1 5PU

UK | UAE | INDIA | SINGAPORE

ISBN 978-93-5551-333-5

www.bpbonline.com

Jai Guru Ji

Dedicated to

My beloved Parents:

Shri A. P. Kukreti

Smt. Sushila Kukreti

&

My wife ***Shilpi***

About the Author

Praveen Kukreti is Technology Leader who is presently working as a Technology consulting manager in a reputed firm. He has a total of 15+ years of rich experience in the IT industry, spanning across Data center, Solution Architecture, Cloud, Presales and Technology consulting. Praveen has been instrumental in providing solutions across various engagements, focused around On-Prem and Google Cloud Platform. He is very passionate about learning new technologies and keeps himself updated on various advancements in the IT industry. In his free time Praveen likes reading, writing and traveling.

About the Reviewer

Sangram Rath is a seasoned technology professional with over 17 years of experience of which over a decade has been on Cloud & related technologies. He works as an independent Cloud Architect & Technology Advisor helping companies, mostly startups & small businesses adopt Cloud and other emerging technologies such as DevOps, Containers etc. He is also the founder of OD10 Ventures, a technology advisory and consulting firm.

He has numerous certifications across many technologies including Microsoft Azure, AWS, Google Cloud Platform & Kubernetes (Linux Foundation). He has trained thousands of professionals on these technologies since 2013 and is a Microsoft Certified Trainer as well.

He has authored the book titled Hybrid Cloud Management using RedHat CloudForms published by Packt and reviewed titles in the past such as Automation Solutions with Chef Automate. Sangram has spoken at conferences such as Open Source Summit (NA & Europe), Digital Ocean Tide, AzConf and has been a hackathon judge too.

He quit the full time corporate world to be able to work on diverse projects and emerging technologies, moved back to his home state of Odisha and is currently based out of Bhubaneswar. He is an art enthusiast, loves food and travelling, especially self-drive road trips.

Acknowledgement

There are a few people I want to thank for the continued and ongoing support they have given me during the writing of this book. First and foremost, I would like to thank the Almighty for His blessings, that have helped me in successfully completing this book. I would like to thank my parents and my wife for continuously encouraging me to write the book — I could have never completed this book without their support.

I am also grateful to my friends who were there to provide required support as and when needed. Their timely inputs have really helped me a lot during this journey. I will specifically like to mention **Mr Ambuj Prakash** and **Mr Jitendra Gupta** for their valuable suggestions.

My gratitude also goes to the team at BPB Publications for being supportive enough to provide me quite a long time to finish the book. You have been really understanding about various concerns that I had during the writing of this book.

Preface

This book primarily focuses on creating a solid foundation for anyone who wants to learn Cloud computing in general and Google Cloud Platform in particular. We have arranged the overall content of this book in 13 chapters. In each chapter, we have tried to explain about the specific subject in a simple language, post which we have gone a little deeper into each and every topic that is part of the subject. As a result readers develop a fair understanding of Google Cloud services and their functionality. Readers will also find various use cases, case studies and lab activities that will help in enhancing their understanding in a more practical way.

There are questions at the end of each chapter that will test your knowledge. You will also find important Key Terms and Links at the end of every chapter. While Key Terms will give you brief one liner explanation of some of the important terminology that has been used in a particular chapter, links will take you to the Google Cloud official documentation links that cover these topics in a detailed fashion.

Since this book has been written considering a beginner in mind, it has been kept as simple as possible. Any learner with some IT background should find it easy to connect with this book. This book will hopefully meet your expectations as it is written with lots of passion. Following are the details of the chapters that are covered in this book:

Chapter 1 will cover cloud computing fundamentals. We talk about various service models used in cloud computing, benefits associated with cloud computing. We also briefly touch upon major cloud service providers available in the industry.

Chapter 2 will cover various compute options available in Google Cloud Platform. We cover various IaaS and serverless options available in Google cloud for running containerized and non-containerized workload. There is a small lab exercise at the end of the chapter that will help readers in getting familiar with GCP console.

Chapter 3 will cover various storage options available in Google cloud. We will discuss briefly about storage encryption available in Google Cloud and

some guidelines to help readers decide the appropriate storage option for their requirement.

Chapter 4 discusses major database services available in GCP. We will see various SQL and No-SQL database services available in GCP. At the end of the chapter, there is a small lab exercise which will help readers in getting some hands-on experience using Google cloud console.

Chapter 5 discusses about various network services available in Google Cloud. We start with explaining Organization structure as it created a basic understanding of creating a landing zone. After this, we discuss briefly about various network services in an easy-to-understand way.

Chapter 6 This chapter covers one of the most important topics in GCP, that is, security and monitoring services in Google Cloud. Here, we discuss various security aspects in Google Cloud such as IAM roles and permissions, service account, container security, Cloud DNS, Firewall and so on. We end the chapter with some security best practice that needs to be followed in GCP.

Chapter 7 touches upon one of the most sought-after field in today's IT world, that is, Big Data. We discuss about various Data Transformation, visualization and analytics services available in Google Cloud Platform. We also discuss about various use cases that help readers in getting a clear understanding of how this various data services work together in order to provide the desired outcome.

Chapter 8 again touches a very important subject, that is, AI/ML. Here we initially talk about Machine Learning fundamentals followed by associated services provided by GCP that help in creating solutions.

Chapter 9 discusses various orchestration services available in Google Cloud Platform. Orchestration is defined as automatic configuration, coordination and execution of computer system and services. GCP provides various tools and services that can be used for orchestration.

Chapter 10 touches a very important topic, that is, migration services in Google Cloud Platform. There are many customers who are looking to migrate their existing workload to the cloud. There are many tools and services available within GCP that can help customer in performing migration assessment, planning and execution. In this chapter, we have discussed about those services.

Chapter 11 touches another important topic, that is, migration services in Google Cloud Platform. There are many customers who are looking to migrate their existing workload to the cloud. There are many tools and services available within GCP that can help customer in performing migration assessment, planning and execution. In this chapter, we have discussed about those services.

Chapter 12 There are many cloud service providers available in the market out of which, GCP is relatively new. GCP keeps on adding some new features and functionalities and hence in this chapter, we have tried covering some of those services that were added some time later.

Chapter 13 covers three use cases that cover areas such as Landing Zone implementation, Data Transformation and Website deployment specific requirements. This will help readers in correlating their already attained knowledge with some practical real world examples.

Coloured Images

Please follow the link to download the
Coloured Images of the book:

https://rebrand.ly/k4vjp9e

We have code bundles from our rich catalogue of books and videos available at **https://github.com/bpbpublications**. Check them out!

Errata

We take immense pride in our work at BPB Publications and follow best practices to ensure the accuracy of our content to provide with an indulging reading experience to our subscribers. Our readers are our mirrors, and we use their inputs to reflect and improve upon human errors, if any, that may have occurred during the publishing processes involved. To let us maintain the quality and help us reach out to any readers who might be having difficulties due to any unforeseen errors, please write to us at :

errata@bpbonline.com

Your support, suggestions and feedbacks are highly appreciated by the BPB Publications' Family.

exclusive discounts and offers on BPB books and eBooks.

Piracy

If you come across any illegal copies of our works in any form on the internet, we would be grateful if you would provide us with the location address or website name. Please contact us at **business@bpbonline.com** with a link to the material.

If you are interested in becoming an author

If there is a topic that you have expertise in, and you are interested in either writing or contributing to a book, please visit **www.bpbonline.com**. We have worked with thousands of developers and tech professionals, just like you, to help them share their insights with the global tech community. You can make a general application, apply for a specific hot topic that we are recruiting an author for, or submit your own idea.

Reviews

Please leave a review. Once you have read and used this book, why not leave a review on the site that you purchased it from? Potential readers can then see and use your unbiased opinion to make purchase decisions. We at BPB can understand what you think about our products, and our authors can see your feedback on their book. Thank you!

For more information about BPB, please visit **www.bpbonline.com**.

Table of Contents

CHAPTER 1

Cloud Computing Fundamentals

Line-of-business leaders everywhere are bypassing IT departments to get applications from the cloud (also known as software as a service, or SaaS) and paying for them like they would a magazine subscription. And when the service is no longer required, they can cancel that subscription with no equipment left unused in the corner."

Daryl Plummer, Gartner Analyst

Introduction

The above mentioned quote statement clearly defines the importance of the Public cloud in the current era of application modernization and cloud transformation. Cloud computing is one of the most important innovations in modern IT that touches each and every part of the technology domain. In this chapter, we will try to understand the basics of cloud computing in a manner that is easy to understand, even for a beginner.

Structure

In this chapter, we will cover the following topics:

- What is cloud computing?
- Various service models in cloud computing
- Various deployment models in cloud computing
- Major cloud service providers in the industry
- OPEX versus CAPEX models in the cloud
- Benefits of cloud computing
- Risks associated with cloud computing
- Introduction to Google Cloud Platform

Objectives

After reading this chapter, you will be able to understand the basics of cloud computing. It will create the foundation of the cloud computing journey. You will also be able to develop a basic understanding of various cloud service providers and will dig a bit deeper into Google Cloud.

What is cloud computing?

Traditionally, Information Technology (IT) has been one of the most dynamic fields that have impacted each and every aspect of our lives. IT services have gone through tremendous changes over the past few years. Looking at the history of IT evolution, we have seen a centralized approach of computing systems like Mainframes to a more decentralized approach of having distributed client-server architecture. Later on, with the introduction of the Storage Area Network and Network Attached Storage, we once again started looking at storing the data at a centralized location. Virtualization has played a very important role in making sure that we optimize resource consumption without compromising on reliability and performance. Traditionally, organizations have been storing data in their in-house data centers that provide all power, cooling, and cabling requirements to host servers, network equipments and storage arrays at a specific location. Then came the co-location model wherein a service provider provides the data center building with all cooling, power, and so on, and companies would host their servers, networking equipment, and storage arrays. Now with the advent of cloud computing, we have come up to a step further in the journey wherein we can see various cloud service providers who provide on-demand availability of computing and other resources. You can simply use these services for a specific duration and may not have to bother with managing or hosting these services. You just pay for what you use and have the option to build it from scratch by yourself.

According to NIST, "*Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models.*"

Various service models in cloud computing

Cloud computing can be categorized into the following three categories:

- **Infrastructure as a service (IaaS)**: In Infrastructure as a service offering, the cloud service provider is responsible for managing underlying infrastructure (Compute, Storage, Network, and so on). The customer is responsible for developing, configuring, and managing the application and associated platforms (OS, runtime environment, DB, and so on). Customer is required to pay for the infrastructure that they are going to use for running the applications for a specified duration.
- **Platform as a service (PaaS)**: Under platform as a service offering, the cloud service provider is responsible for providing and maintaining a run time environment for running the application along with the underlying infrastructure. The customer is responsible for developing, deploying, and managing the applications, and all other things are taken care of by the cloud service provider. Google App Engine, AWS Elastic Beanstalk, and so on are some examples of platform-as-a-service offerings.
- **Software as a service (SaaS)**: Under software as a service offering, the cloud service provider is responsible for providing and managing the software and it is dependencies. The customer is just going to pay for the software for the duration he is going to use this software. Under SaaS, everything is managed by the software vendor or the cloud service provider, including application, runtime environment, OS infrastructure, patching, and so on.

Cloud models comparison

Table 1.1 compares the various computing models:

Infrastructure as a service(IaaS)	Platform as a service(PaaS)	Software as a service(SaaS)
Application	Application	Application
Data	Data	Data
Runtime environment	Runtime environment	Runtime environment
Middleware	Middleware	Middleware
OS	OS	OS

Infrastructure as a service(IaaS)	Platform as a service(PaaS)	Software as a service(SaaS)
Virtualization	Virtualization	Virtualization
Server	Server	Server
Storage	Storage	Storage
Networking	Networking	Networking
		Managed by the customer
		Managed by cloud service providers

Table 1.1: *Cloud model comparison*

Various deployment models in cloud computing

Deployment models are related to the ownership, hosting environment, and other specifications related to deployment infrastructure and associated services. Considering previous things in mind, cloud deployment models can broadly be categorized under the following categories:

- **Public cloud:** Public cloud is a type of deployment model in which all the cloud hosting services are deployed and managed at the cloud service provider's data center, and these services are open to being used by all users. This cloud deployment model makes it possible for anybody to use cloud services. Organizations with resource requirements that fluctuate with demand, are unpredictable, and have relatively low-security requirements can make use of this deployment model. In this deployment model, all software, hardware, and associated facilities are maintained and managed by a cloud service provider.
- **Private cloud**: In this deployment model, cloud services are hosted for a single organization. In this, these services are managed and maintained by either the organization itself or by any third-party provider. Organizations that want to have granular control over their entire environment and have security-related concerns are the ones that prefer this model. Nowadays, there are many cloud service providers who have come up with this kind of deployment model where the entire environment is dedicated to a single customer and is not shared among other organizations. Sometimes cloud services are deployed at the

organization's data center itself or at some third-party-provided co-location facility.

- **Community cloud**: This type of deployment model is similar to the Private cloud. The only difference is that in this type of model, the environment is dedicated to more than one customer who share the same concerns or are part of the same community. Most of the time, the community itself is responsible for owning, maintaining, and managing the community cloud; however, there are scenarios where a cloud service provider is owning, managing, and maintaining the cloud environment. The community cloud is a little difficult to manage as the data of various customers that are part of the same community resides in the same data center.
- **Hybrid cloud**: Hybrid cloud is a combination of two or more deployment models. This model is considered when any organization wants to make use of benefits associated with multiple deployment models. For example, for some applications, organizations may have security concerns, and for a certain set of applications, they do not have very rigid security and compliance-specific concerns. In this type of scenario, organizations may decide to have a combination of Public and Private cloud. This deployment approach comes up with the additional requirement of managing both the cloud environment using a single platform.

Figure 1.1 features the various deployment models:

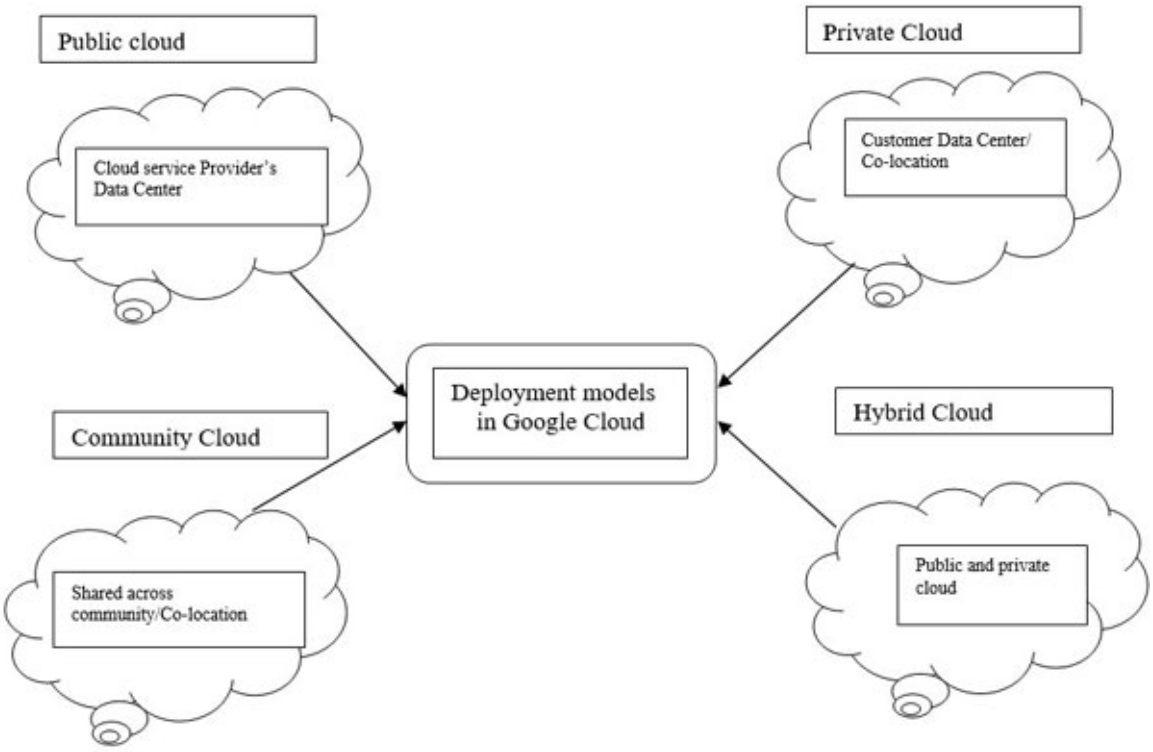

Figure 1.1: *Various deployment models*

Major cloud service providers in the industry

Even though, at present, there are many cloud service providers in the industry, the majority of the cloud market are being captured by the following three cloud service providers:

- **Amazon Web Services** (**AWS**): Amazon Web Services is one of the oldest and most widely used cloud service providers in the industry and part of Amazon Inc. AWS is considered as one of the most popular and widely adopted cloud computing platforms that spans across many geographical locations around the globe. As of date, AWS is spread across 26 regions across the globe and is continuously increasing. This platform is managed by an AWS console, using which various applications and services can be created and deployed. We will cover regions and zones in a later part of this book.
- **Microsoft Azure**: Microsoft Azure is another major cloud service provider in the industry owned and managed by Microsoft. This cloud service provider is also very popular among users. Many organizations have their workloads running on Microsoft Azure. At present, Azure is

available in 60+ regions around the world. Microsoft Azure is considered a preferred choice for running Microsoft-based applications. This cloud service platform is also available in the market since long and continuously expanding.

- **Google Cloud Platform** (**GCP**): Google Cloud Platform is relatively new but considered one of the fastest growing cloud services platforms in the industry. As of date, it has around 35 regions spread across the globe that are increasing at a rapid rate. This platform utilizes the same infrastructure that is used to run their existing Google services, such as Google Workspace (Previously called G Suite), YouTube, Gmail, Google maps, and so on. GCP uses its undersea network cables for network communication. As you know, the book focuses on the Google Cloud Platform, and we will be covering all major services that are part of the Google Cloud Platform in the upcoming chapters.

OPEX versus CAPEX models

Traditionally, companies have been following the CAPEX model for their IT workload. CAPEX stands for capital expenditure. In this model, the customer makes a one-time investment to build infrastructure inside a data centers where they run their workload. Usually, this one-time investment is high and also lacks any future growth and scalability requirement for their workload. Apart from it, in case if the company shuts down its operations, this investment often gets wasted. Assigning and up-skilling manpower is another problem associated with this model.

Cloud has enabled organizations to make use of the OPEX model instead of a CAPEX model. OPEX stands for operational expenditure. In the OPEX model, there is no upfront investment for procuring the resources as this model works on a pay as you go model. As mentioned previously, IT organizations can opt for IaaS, PaaS, or SaaS-based models depending upon their needs, which helps in reducing the upfront cost of procuring the resources.

Benefits of cloud computing

Cloud computing has many benefits associated with it. Some of the most important benefits can be categorized as follows:

- **Scalability**: Scalability is one of the major benefits of cloud computing. You can scale resources on the go. Auto-scalability is one of the major

benefits of cloud computing, where resources can automatically scale up as the workload grows.

- **No upfront investment for infra procurement**: Another major benefit of cloud computing is that it works on a pay-as-you-go model. It means that there is no need to procure the infrastructure in one go. In the pay-as-you go model, there are various options available to choose from, such as sustained use discount, committed use discount, and so on. Customers can choose from any of the options as per their requirements. We will discuss it in subsequent chapters.
- **Fast to market**: Developing and deploying applications on the cloud help in speeding up the process and bringing applications into production. Cloud enables us to provision resources much faster, which speeds up the overall process.
- **Patch updates**: Cloud service providers take care of updating any kind of patch updates that are required. Customers need not to worry about it.
- **Ease of operation and maintenance**: One of the major benefits associated with the cloud-based model is the ease of operation and maintenance. Cloud service providers provide you with various options to choose from, that is, IaaS, PaaS, or SaaS. Depending upon the model you choose, operations and maintenance fall in the kitty of the cloud service provider.
- **Ease of Exit**: It is much easier for the companies to exit if they are closing their business as there was no initial DC setup involved. Companies can terminate their contracts with their respective cloud service providers if any such need arises.

Risks associated with cloud computing

At present, cloud service providers are providing many features and functionalities that are helping organizations to adopt the cloud model and reduce the risks associated. However, there can be a few risk areas, such as:

- **Compliance-specific risk**: Many organizations want their workload to be running within a specific geographical location or country, as there may be country-specific laws that prohibits them to move their data outside their country. In this case, it may become difficult for the organizations to move their workload into the cloud as cloud CSP may not have its presence in a specific geographical location.

- **No fine-grain control over resources**: Once customers have chosen a specific deployment model, they may lose control over resources and services. In case there are some configuration changes required at a later stage, they may face problems depending upon the model that they have chosen.
- **Skillset challenges**: When a customer is moving their workload to the cloud, they need to either upskill their existing resources or need to hire new people that may be difficult to find. Sometimes this transition is very fast, which leaves some gaps in between.
- **Migration risks**: While migrating workload to the cloud, there may be some applications running on the legacy environment, and it may be difficult to move those applications to the cloud. Sometimes you may need to make major changes in the application code to make them cloud-native. This requires tremendous time and effort and can delay migration. Sometimes there can be applications that are difficult to move to the cloud.
- **Security risks**: In cloud computing, data majorly resides in cloud service providers' data centers and may be vulnerable to risks. Even though there are many security features available in the cloud, how to implement them is one of the major requirements we have. Setting adequate firewall rules, assigning the right identity and access control rules, and so on require due diligence and effort.

Introduction to Google Cloud Platform

As stated earlier, Google Cloud Platform is one of the fastest-growing cloud service providers in the market. Historically, Google has been doing all kinds of innovations in the technology world. For example, Kubernetes, which is a widely popular orchestration engine used to run containerized workloads both on-prem and on the cloud, was originally developed at Google. Later on, this was made Open-Source by Google. Now Kubernetes engine is a major offering of all cloud service providers' portfolios.

Google cloud platform has various offerings under IaaS, PaaS, SaaS, and CaaS categories. We will be discussing about these offerings in the subsequent chapters.

Regions and Zones: Google Cloud Platform is distributed across various regions and Zones. Regions can be described as a specific geographical area or

location where GCP resources are made available. This Geographical area is divided into three or more zones. There are some resources that reside in a region and are called regional resources; similarly, there are some resources that reside in a zone that are called zonal resources. At present, Google has 35 regions and 106 zones, and this number is continuously increasing. We will be discussing more about regions and zones in subsequent chapters.

These geographically distributed regions are mostly connected with each other through undersea cables. These cables are the same that carry Google's GSuite workload (that is, Gmail, YouTube, and so on).

Some examples of regions and zones can be seen in *Table 1.2*:

Region Name	Zones
us-west1	• us-west1-a • us-west1-b • us-west1-c
us-west2	• us-west2-a • us-west2-b • us-west2-c
us-west3	• us-west3-a • us-west3-b • us-west3-c
us-west4	• us-west4-a • us-west4-b • us-west4-c
us-central1	• us-central1-a • us-central1-b • us-central1-c • us-central1-f
us-east1	• us-east1-b • us-east1-c • us-east1-d

Region Name	Zones
us-east4	• us-east4-a • us-east4-b • us-east4-c
northamerica-northeast1	• northamerica-northeast1-a • northamerica-northeast1-b • northamerica-northeast1-c
southamerica-east1	• southamerica-east1-a • southamerica-east1-b • southamerica-east1-c
southamerica-west1	• southamerica-west1-a • southamerica-west1-b • southamerica-west1-c
europe-west1	• europe-west1-b • europe-west1-c • europe-west1-d
europe-west2	• europe-west2-a • europe-west2-b • europe-west2-c

Table 1.2: *Current regions and zones in Google Cloud Platform*

Google focuses a lot on research and innovation and keeps on coming up with additional services that are part of their IaaS, PaaS, and SaaS offerings.

Google Cloud Project

Whatever you create in Google Cloud is part of a project. So before creating any resource in Google Cloud you must create a project. Whatever resources that you create in a project share the same network and hence can have easy interaction with other resources. A project is identified by the following three:

- **Project Name**
- **Project ID**
- **Project Number**

Project ID is always unique in GCP. It can either be created by a user or can be provided by Google.

Different ways of accessing Google cloud platform

There are three different ways using which you can access Google Cloud Platform.

- **Command Line Interface**: One way to interact with the Google cloud platform is through the command line interface. You use command line tools such as gcloud, gsutil, bq, and kubectl to interact with Google cloud resources that are part of Google Cloud SDK. You can use the command line either by installing Google cloud CLI on your system or can use browser-based Cloud Shell, available in Google Cloud Console. Refer to *figure 1.2*:

Figure 1.2: *Command line interface*
Source: *https://cloud.google.com/shell?hl=es*

- **Google Cloud Console**: Google cloud console is another way of interacting with Google cloud. It is a graphical user interface that is used to create or manage various resources in the Google cloud platform. You begin with creating a project, and under this project, you can create resources. You can always create multiple projects within Google Cloud and enable billing for them at the project level. Refer to *figure 1.3*:

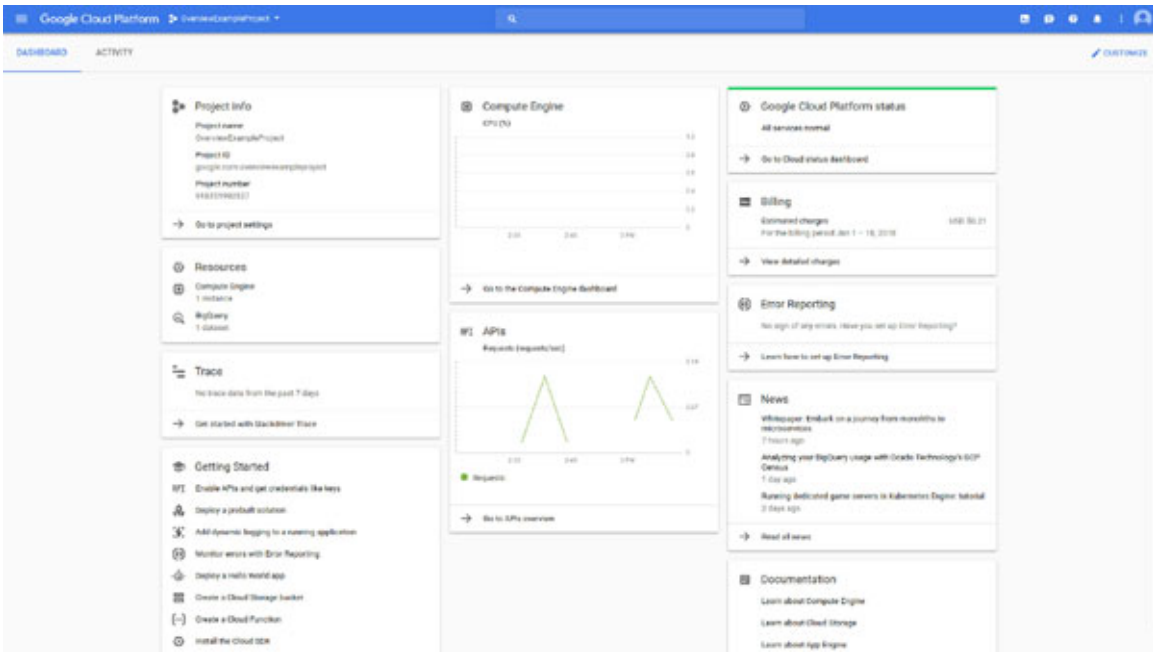

Figure 1.3: *Google Cloud Console*
Source: *https://cloud.google.com/docs/overview*

- **Through APIs**: Another way to interact with Google Cloud is through APIs. This is a set of client libraries that can be used to access various products in Google cloud. These APIs can be integrated with other applications through these libraries. The following is the link to Google cloud APIs:

  https://cloud.google.com/apis

Conclusion

Cloud computing is one of the most demanding skills in the market, which is going through innovation each and every passing day. It does not matter whether you are from a development background or a system administrator; cloud computing skills are a must in the present world of information technology.

Key terms

- **NSIT:** National school of standard and technology
- **IaaS:** Infrastructure as a service
- **PaaS:** Platform as a service

- **SaaS:** Software as a service
- **Opex:** Operational expenditure
- **Capex:** Capital expenditure
- **GCP:** Google Cloud Platform
- **AWS:** Amazon Web services

Questions

1. What is cloud computing?
2. What are various service models in cloud computing?
3. What are various deployment models in cloud computing?
4. What are the benefits and risks that you have in cloud computing?
5. How do you differentiate between OPEX and CAPEX model?
6. What is Google cloud Platform?
7. How many regions and zones that you have in the Google Cloud Platform?

Further reading

- **https://cloud.google.com/learn/what-is-cloud-computing**

CHAPTER 2

Compute in Google Cloud

Introduction

Compute is one of the most important parts of any data center because this is where you run your applications. Computing power in any kind of data center plays a very important role in the functioning of any application. When we mention compute, we are actually mentioning about the infrastructure and services on which you run your applications. You can use Bare Metal Servers, Virtual Machines, and Kubernetes Engines as part of your compute. You deploy and run your applications on these compute resources. These are the key resources using which you perform computation activities and include memory, processing power, and so on.

Structure

In this chapter, we will cover the following topics:

- Google Cloud Compute Engine
- Introduction to Kubernetes Cluster
- Google Kubernetes Engine(GKE)
- Cloud Run
- Google App Engine
- Cloud Functions
- Cost comparison between various compute option
- Choosing the correct compute option in GCP
- Lab Exercise

Objectives

In this chapter, we are going to discuss about various compute options available in GCP. We are going to mention various IaaS and PaaS offerings. After going through this chapter, a beginner must be able to select an

appropriate compute option for his requirements. **We advise learners to go through the Google Cloud document links in order to keep themselves updated with the frequent changes to these services**.

Google Cloud Compute Engine

Google Cloud Compute Engine gives us various options to choose from. Compute instances are the virtual machines where you run your application workload. When you are considering Infrastructure-as-a-Service as one of the options for your workload, you can select from various options provided by the cloud service providers. The types of Compute Engines in GCP are discussed as follows.

Predefined machine types

Predefined machine types are the machine types that are already preconfigured and available to be used by the customers. Customers have the choice to select among the available options that meet their requirements. Let us have a look at these categories in order to understand them better.

Machine families are defined to categorize various instances as per workload requirements. Customers can select any of the predefined machine types under specific categories that meet their needs. Compute Engine families can broadly be categorized as follows:

- **General purpose**: Balanced approach of compute and cost for generic computing needs.
- **Compute optimized**: For high-performance needs for running a workload.
- **Memory optimized**: For running memory-intensive workload.
- **Accelerator optimized**: Used for running massive compute requirements (parallel processing) for specific use cases such as **High-Performance Compute** (**HPC**) or GPU and so on.

Let us now dig a bit deeper into each family type:

General purpose: When you want the best price-to-performance ratio, general-purpose machine types are the most suitable ones. The following are the various categories under General purpose machine types:

- **E2 machine series**: This series can have up to 32vCPU and 128 GB of memory. This series is considered for having compute resources at the lowest cost with a committed use discount. It uses Intel or a second-generation AMD EPYC Rome processor.
- **N1 machine series**: This series can have up to 96vCPU with 6.5 GB of memory per virtual CPU. It uses Intel Sandy Bridge, Ivy Bridge, Haswell, Broadwell, and Skylake processors.
- **N2 machine series**: This series can have up to 128 vCPU with 8 GB of memory per virtual CPU. It uses Intel Ice Lake and Cascade Lake CPU.
- **N2D Series machine**: This series can have up to 224 vCPUs, and 8 GB of memory per virtual CPU. It uses AMD EPYC Rome and third-generation AMD EPYC Milan processors.
- **Tau T2D machine series**: This series can have up to 60 vCPUs, and 4 GB of memory per virtual CPU. It uses AMD EPYC Milan processors. Since it does not have cluster threading enabled on it, each virtual CPU represents a physical core.

Compute optimized: This machine type is considered suitable for running compute-intensive workloads where high performance is desired. It uses Intel Scalable Processor (Cascade Lake) and AMD EPYC Milan processor. The following are the various categories under compute-optimized machine types:

- **C2 machine series**: This series can have up to 60 vCPUs, and 4 GB of memory per virtual CPU. It uses Intel Scalable (Cascade Lake) processors.
- **C2D machine series**: This series can have up to 112 vCPUs, and 4 GB of memory per virtual CPU. It uses third-generation AMD EPYC Milan processors.

Memory optimized: This machine type is considered suitable for memory-intensive workloads. This series offers more memory per core. You can have up to 12 TB of memory in this series.

Accelerator optimized: This machine type is used for high-performance computing and machine learning kind of requirements where you have massive parallelized processing requirements.

Custom machine types

When any of the predefined machine types are unable to fulfill your requirement, then you select the customer machine type. Here, you get an option to select the configuration as per your requirement (Compute, memory, and so on). You also get an option to create a template for this custom machine and use it at a later stage.

Google compute engine provisioning models

Google compute engine also uses various provisioning models. The following are the various provisioning models offered by GCE:

Standard VM instance: Standard VMs are the ones over which the end-user has full control with respect to the running time. While configuring these VMs user defines for how long that machine needs to run. Either the user can choose the "*pay as you go*" model or can request it to run for a specific time (Committed model).

Preemptible VM instance: Preemptible VMs are the VMs that can be terminated by GCP within 24 hours of their creation. These VMs are used in scenarios where your requirement is to run VMs for a specific time to finish a specific job or task. Preemptible instances can be terminated at any time; hence, they should be used in scenarios in which stopping these jobs does not impact your applications. Preemptible VMs will stop automatically after 24 hours. You end up saving significant costs (60%–90%) by using preemptible VMs.

Spot VM instance: GCP Spot VM instances provide you an option to run your jobs for more than 24 hours and also provides significant discount; however, like preemptible VMs, they can also get terminated anytime by Google. They also come with a 60%–91% discount. It is recommended to use a Spot instance when stopping a virtual machine is not going to impact your application. If you want to run batch jobs for a specific time period, then you can make use of spot instances.

*Preemptible and Spot VMs cannot live to migrate to Standard VMs.

Introduction to Kubernetes cluster

A Kubernetes cluster can be defined as a combination of various nodes that are used to run a containerized workload or application and manage it. It can be defined as an orchestration system that is used to run and manage containers.

In order to understand a Kubernetes cluster, we need to understand the concept of containerization.

Microservices

Traditionally, we have seen that the applications have been developed using a monolith approach, that is, a single application having various functions performing different tasks that have very less dependencies between each other. For example, a single organization portal having Finance, Human resource, and Marketing functions as part of a single application. This application designing and development approach has its challenges associated with it. For example, if one function has some issues in it, the overall application gets impacted, or in case if you want to add some specific features to a single function, the overall application needs to get rebuild or redeployed.

In a microservices-based architecture, a single application is subdivided into separate microservices that are loosely coupled, that is, they have very less dependencies between each other. You package application code for the microservice, and its dependencies together are in the form of a container image. These container images run only when there is some sort of incoming request.

Container

Containers are the packages that have application code and dependencies to run that application. Dependencies include the application runtime environment and runtime libraries that are required to run it. Containers do not require the full operating system to run.

Traditionally, we have seen an application running directly on top of an operating system installed on a hardware machine or on top of a virtual machine. Containers do not require an entire operating system to run and, unlike virtual machines, use virtualization at the OS level. CPU and memory virtualization is done at the OS level, which makes container images to run faster. Since you have only application code and its dependencies, it also makes the container smaller and portable.

Running a single microservice directly on top of an operating system is not an attractive choice, as it results in wasting compute resources. Apart from it, running multiple applications directly on a single VM also has its challenges, such as conflicting libraries and performance issues. Containers are best suited

for a microservices-based design approach since we can run multiple microservices in the form of containers that have their application code and dependencies packaged together. We take advantage of virtualization, which results in the optimum use of computer resources within minimum performance bottlenecks. Each microservice has its own runtime environment packaged together that makes it portable.

POD

A POD is a single, deployable unit inside a Kubernetes cluster that consists of one or more container images. It is a group of one or more containers that share a common network namespace. PODs are used to run multiple processes in the form of separate containers that need to run together. These processes or microservices may require to have some sort of communication with each other. PODs can also specify some shared storage that can be shared among containers within a single POD.

Kubernetes cluster

Figure 2.1 features the architecture of a Kubernetes Cluster:

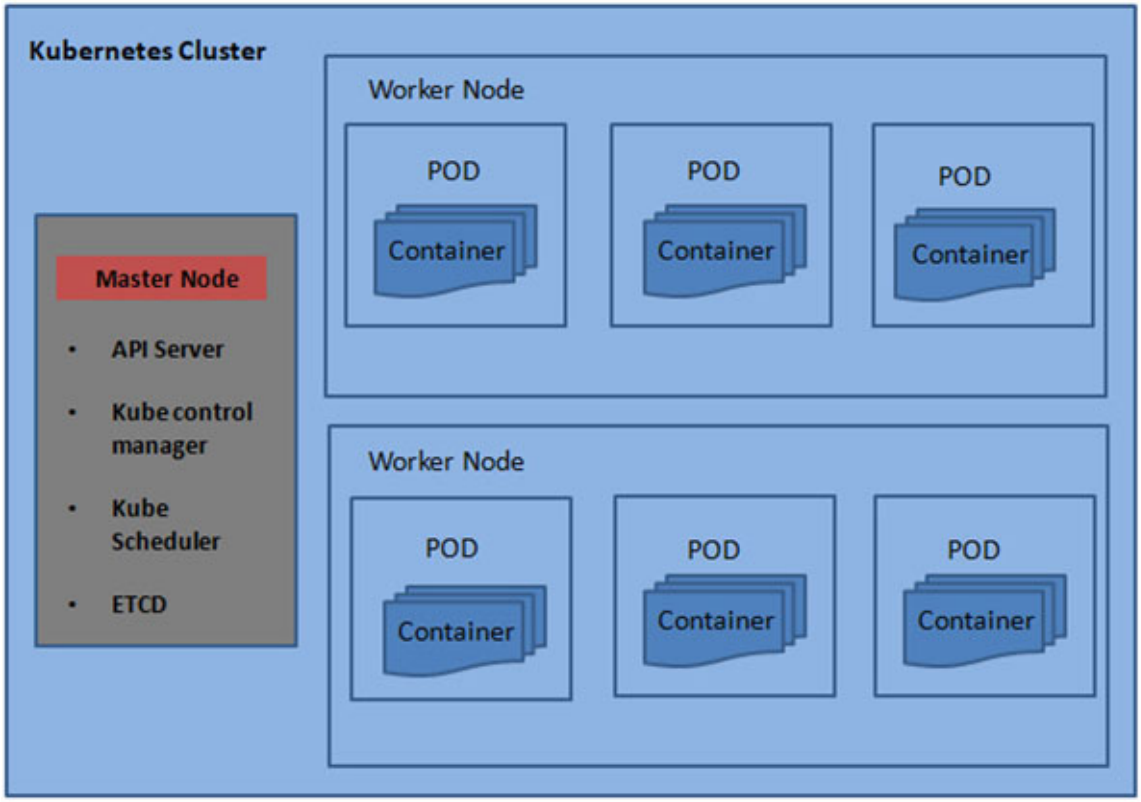

Figure 2.1: *Architecture of a Kubernetes Cluster*

A Kubernetes cluster can have the following two types of nodes:

- Master Node
- Worker Node

Master Node

The master node is responsible for controlling and managing the cluster. This is where your services run that is responsible for managing and scheduling the loads running inside a Kubernetes cluster. The following are the services that run inside a Kubernetes cluster:

- **kube-apiserver**: kube-apiserver is responsible for exposing the Kubernetes APIs. It is the gateway for the Kubernetes cluster. It authenticates and validates all API requests and also updates the etcd database when a request is complete. It also reads the data from etcd database in case if there is a read request. Users can communicate with kube-apiserver using kubectl command. You can make get, put, list, and delete requests for a resource inside a Kubernetes cluster through kube-apiserver.
- **etcd**: etcd is where you store information about all cluster data. It is a key value store where configuration data, metadata and state data for a Kubernetes cluster is stored. As mentioned earlier kube-api server interacts and updates etcd. etcd keeps on getting updates as per any post or delete request made through kube-apiserver.
- **kube-controller-manager**: kube-controller-manager monitors the state of the Kubernetes cluster. It has control loops running in it. The Kube controller manager makes the change request and tries to move the cluster from the current state to the desired state. It continuously interacts with the kube-apiserver to get the state of the nodes. There are various types of controllers, such as node controller, job controller, deployment controller, and replica controller.
- **kube-scheduler**: kube-scheduler is responsible for monitoring the state of the PODs and restarts them on other nodes if the need arises. It also monitors the newly created PODs and assigns them to a node. PODs are where you have one or more container images running.

Worker Node

These are the ones where you have applications/services running in the form of containers inside the PODs. A node can have one or more PODs inside it. Each Kubernetes cluster has one or more worker nodes. The following are node components that run on every worker node:

- **kubelet**: kubelet is an agent that runs inside each worker node of a Kubernetes cluster. It is responsible for making sure that PODs are up and running. Kubelet interacts with the kube-apiserver, and as per the inputs/commands received, it tells the container runtime environment to stop and start the containers.
- **kube-proxy**: kube-proxy runs on each worker node. It is a network-proxy and is responsible for making sure that network rules are intact. It enables network communication between a node and POD.
- **Container runtime**: Container runtime provides the required environment for running containers.

*** Note: Here, we have briefly touched upon the architecture of a Kubernetes cluster so that readers are able to develop a basic understanding of Kubernetes-specific services in the Google Cloud Platform. Kubernetes itself is a broad topic, and in order to develop further understanding, readers are suggested further learning.**

Google Kubernetes Engine

Almost all major cloud providers provide a managed environment for running containerized workloads. This workload runs on Kubernetes Engine. In Google Cloud, this managed offering is called the Google Kubernetes Engine. Traditionally, in an On-Prem environment creating and managing a Kubernetes cluster has been a very tedious task that required specific skillsets that were hard to be found. Google Kubernetes Engine makes it easy for the operations team to create and manage Kubernetes cluster for running containerized workload. In a Google Kubernetes Engine environment, you have multiple compute instances that are grouped together for running containers. In GKE master node is called the “control plane”.

Using the Google Kubernetes Engine for running containerized applications has many additional benefits associated with it that make it easy to manage a cluster, such as auto-scaling, auto node repair, auto upgrades, monitoring load

balancing, and so on. Containerized images can be stored in a Container or Artifact Registry from where this containerized image can be deployed in GKE.

Modes in Google Kubernetes Engine:

GKE has the following two modes of operation:

- **GKE Standard Mode:** In case if you want to have more control over managing a Kubernetes cluster, then you must select GKE standard mode. In this mode, we have node configuration flexibility along with control over managing a Kubernetes cluster. Using this, you get more control over the pricing aspect as you pay for the configuration that you have selected for a Cluster. In scenarios where you have high integration requirements, you should opt for GKE standard mode. The cluster will scale in and scale out according to the configuration parameter that has been set, so node and POD auto-scaling are managed by the customer. Herein, you share the details like the minimum and the maximum number of nodes that are needed. GKE Standards can be regional or zonal.
- **GKE Autopilot Mode:** GKE Autopilot Mode gives you the option of having less overhead in managing a Kubernetes cluster. It this, cluster infrastructure and configuration are managed by Google Cloud. You get a preconfigured cluster with auto-scaling capability, which can scale in and scale out as the load increases or decreases. You pay for the resources that have been consumed, and since it is auto-configured, you do not have much control over the pricing piece. So you end up paying for the storage, memory, and CPU that is consumed while PODs are up and running. Here you just mention the resource requirements, such as processing power, memory, and storage, and GKE takes care of assigning the required number of nodes for running the workload. GKE Autopilot is regional. Here, node auto-scaling is managed by the Google cloud platform, whereas POD horizontal and vertical scaling are managed by the customer himself.

The following link provides you with a comparison between Standard and Autopilot modes:

https://cloud.google.com/kubernetes-engine/docs/concepts/autopilot-overview

Difference between single zone, multi-zone, and regional cluster

A single zone cluster is one where Master and Worker Nodes reside inside a single zone itself. It means that it does not fulfill high availability requirements. If a zone goes down, then the whole cluster will go down. Also, if one Master Node fails, then it impacts the overall cluster.

A multi-zone cluster is one wherein you have Worker Nodes spread across multiple zones, which means PODs are distributed across various nodes across multiple zones, but your Master Node still resides in a specific zone.

A regional cluster is something wherein you have Master and Worker Nodes both residing in multiple zones. It means if a specific zone goes down, PODs running on other nodes will still be running, and since you have multiple master replicas also running across multiple zones, that also does not get impacted.

Cloud Run

In case if you want to abstract away the entire cluster management task, Cloud Run is the option. Cloud Run is a serverless service available in the Google Cloud Platform. In this option, you simply give the container image and use the cloud run command to run this container on the underlying infrastructure, which is Kubernetes cluster. Here PODs and nodes management are taken away from the customer. In Cloud Run, the cluster can scale out instantly from zero as the load increases. Cloud pricing is charged for the resources consumed starting from one-tenth of a second. Google Cloud Run service is regional, so the workload gets automatically replicated across other zones. Cloud Run is used only for running stateless applications. Cloud Run supports programs written in any programming language.

Google App Engine

Google App Engine is a platform-as-a-service offering of the Google Cloud Platform. Google App Engine provides a Web hosting platform to application developers for their Web applications.

It allows developers to solely focus on their application development and not worry about underlying platforms or servers. Google Cloud takes care of it. Developers can build scalable solutions. It has various supporting languages

and tools. Some of the languages that are offered and supported by the Google App Engine are Python, PHP, .NET, Java, Ruby, C#, Go, and Node.Js. Google App Engine also ensures that your application is fully secure and prevents it from external attacks.

The benefits of using an App engine are as follows:

- **High availability**: The App Engine is a regional service. That means if one zone within a region is down, then the application is available in another zone.
- **Faster time to market**: Using an App Engine, you need not to worry about configuring and managing the underlying platform and infrastructure, and developers can solely focus on development activity. This results in saving lots of time to bring the application to production.
- **Scalability**: Google App Engine is scalable. It means when the load on the application increases, it scales up automatically.
- **Reduced cost**: While using App Engine, infrastructure/platform provisioning and management are taken care of by Google Cloud Platform; therefore, customer need not to worry about allocating resources for provisioning and managing the underlying environment.
- **Multiple APIs to use from:** Google Cloud App Engine provides various APIs that can be utilized to add features in an application.

Types of App Engine environment

The types of App Engine environments are described as follows.

App Engine standard environment

The standard environment provides you preconfigured set of the virtual machine to run your applications. Since these machines are preconfigured, it provides you with limitations with respect to the runtime environment that it supports. You can deploy your applications very fast. Applications run on a sandbox environment that supports the following programming languages: Python, Java, Node.js, PHP, Ruby, and Go.

App Engine standard environment is preferred for applications that require rapid scalability due to sudden spikes in traffic.

The advantages are as follows:

- It can scale down to zero when the application is not running.
- Applications can be deployed quickly (within seconds).
- Auto-scaling is very fast. A good option for applications with the unpredictable workload.

The disadvantages are as follows:

- You cannot SSH to the virtual machines where applications are running.
- Very specific programming languages are supported.
- You do not have control over configuring the VMs on which applications are running.
- Runtime flexibility is not available in the standard environment.

App Engine flexible environment

In the App Engine, flexible environment applications run in a Docker container running inside a virtual machine. Since the application code is packaged inside a Docker container, you have a variety of runtime environment to choose from. Herein, you get an option to configure your virtual machines. While applications can be up and running instantly in a standard environment, in a flexible environment, applications take time to come up (a few minutes).

The advantages are as follows:

- It supports multiple programming languages.
- You can SSH to virtual machines and can make configuration changes.

The disadvantages are as follows:

- It cannot scale to zero when the application is not running.
- It takes time to deploy.
- Auto-scaling is not that fast.
- It is costly when you compare it with the App Engine standard environment.

Cloud Functions

Cloud Functions are serverless offering that is triggered in response to an event. It is written in Java, Python, or Go and can get attached to an event. Whenever that event occurs cloud Function can be triggered to perform the

desired task. For example, whenever some files get uploaded in Cloud Storage, a Cloud Function event can be triggered to send a notification.

In Cloud Functions, customers are charged for the time for which the function is running. One of the major benefits of using cloud Functions is that it removes the additional overhead of creating, managing or maintaining the VMs.

Cloud Functions have many use cases, for example, if you want to create an end-to-end ETL data pipelined for analytics, you can trigger a cloud Function as soon as some data gets copied into Cloud Storage, or you can also start any application build that can be triggered through a Cloud Function. We will learn about data pipeline concepts in the later part of this book. So, in summary, Cloud Functions can be used to perform a specific task in response to a specific event. Cloud Function will just perform that task for a specific time. Required underlying resources get created automatically when the Cloud Function executes. Cloud Functions come under **Function-as-a-Service** (**FaaS**) offering.

Cost comparison between various compute options

When we do a cost comparison of various GCP services, we may not have a straightforward answer to it. Different Compute options have different use cases, and depending upon that, we finalize the appropriate option. So, it is better if we have an understanding of those use cases first and then finalizes the compute option for our requirement. Primarily, we have the following options available under compute:

- **Pay-as-you-go model**: You pay for the period for which you have used the resource.
- **Sustained use discount**: This type of discount is given when you use compute resources for a specific time period.
- **Committed use discount**: Committed use discounts are given for using some specific percentage of the resources for a reasonable portion of a month.

Let us have a closer look at the use cases for various compute options and the pricing model associated with them:

Compute Engine: Google Cloud provides us with various machine types to choose from, and the pricing differs as per the configuration requirement. You

will choose to Compute Engine as an option when you want to have fine-grained control over the underlying infrastructure and also adequate skillets available within your implementation and operations team. In case if you are ok with investing in recruiting new resources or up-skilling the existing resources, this is the preferred option. Google Cloud provides you the discounts on the usage of Compute Engine resources in the following ways:

- **Sustained use discount**: When you use memory and compute resources for more than 25% of a month, you automatically get a discount for every minute being used.
    - For N2, N2D, customer machine types, compute optimized: Up to 20%.
    - For general purpose N1 and predefined custom machine types, sole-tenant nodes: Up to 30%.
- **Committed use discount**: Committed use discount is ideal for a predictable workload for a definite timeline. When you are sure of the committed use of compute resources, you select this option. Committed use comes up with one year and three years options
    - For CPU-optimized machine types and GPUs: Up to 57%.
    - For general purpose N1 and predefined custom machine types, sole-tenant nodes: Up to 30%.
- **Google Kubernetes Engine**: As we understand, we have Google Kubernetes Engine in two modes. Following is high level pricing calculation for standard and autopilot modes:
    - **Standard mode**: In standard mode, a cluster management fee of 0.10$ is charged for individual Kubernetes cluster after the free tier. Apart from this, the Google Kubernetes Engine charges for the Worker nodes that have been allocated to the Kubernetes cluster. These nodes are charged for every one second with a minimal 1 minute of usage. Worker nodes have the same committed and sustained use discount applicable as for compute engine.
    - **Autopilot mode**: The cluster management fee for Autopilot mode is the same as for the Standard mode; however, here, compute resources (memory, storage, and CPU) that are being used by the scheduled PODs.

Please check the following link to get further details about GKE pricing: **https://cloud.google.com/kubernetes-engine/pricing/**

- **Cloud Run**: Cloud Run pricing is calculated for the amount of memory, CPU, and networking resources being consumed. Initially, there is a free tier available, post which users are charged for as per the resources being consumed. For example, as ofdate, 180,000 vCPU seconds are free per month, post which users are charged per vCPU second. Check the following link to get further details on the cloud run pricing: **https://cloud.google.com/run/pricing**
- **App Engine**: App Engine pricing differs for the standard and flexible environment. There is a free tier available for App Engine standard environment, post which users are charged on per hourly basis for different classes. Different App Engine instance classes have different pricing associated with them. These classes differ according to the CPU and memory consumption requirement and also according to the manual and automatic way of scaling. Additionally, you are charged for Persistent Storage consumed. App Engine's flexible environment does not have any free tier available. Also, there are no predefined classes here. Users are charged according to the CPU and memory consumed by the application. VM resources are billed per core per hour and per GB per hour. Apart from it, there is network cost involved for outbound/egress traffic. There is no incoming network traffic cost in the Google Cloud Platform. Additionally, you are charged for the Persistent Storage that you have consumed. Check the following link for more details on App Engine pricing: **https://cloud.google.com/appengine/pricing/**
- **Cloud Functions**: Cloud Function pricing is a combination of various factors. There is a free tier available post which you are charged:
    - You are charged as per the number of invocations by Cloud Function. Examples of these invocations are HTTP calls or API calls made by Cloud Function. You are charged for per million invocations.
    - You are also charged for the CPU and memory consumed by the Cloud Function while it is running. For Cloud Functions in GCP, it is measured in the form of compute time. This compute time is measured in GB-Seconds and GHz-Seconds and has some fixed pricing associated with it.

- If the minimum number of Cloud Function instances is set, then you are also charged for idle instances.
- You are also charged for outbound data, and as always, there are no charges for inbound data.
- Check the following link for more details on cloud function pricing: **https://cloud.google.com/functions/pricing**

Choosing the correct compute option in GCP

Choosing the right compute option is very important while designing your architecture and requires knowledge and extensive research on available Google Cloud Compute Services. When you are looking for compute options, you divide the requirement into the following two categories:

Infrastructure-as-a-Service (IaaS): You choose the Infrastructure-as-a-Service option when you want to have a granular level control on the underlying platform and configuration of the compute that you are planning to use. If you want to have full control over the underlying platform, processor, memory, and other configurations, you can opt for IaaS offering like compute engine. You will primarily select this option to run any monolithic application.

Container-as-a-Service (CaaS): CaaS offerings are considered when you consider running your containerized workload on the Google Cloud Platform and want to have some control on the underlying platform and Kubernetes cluster. Selecting a GKE is a better choice when you are running licensed software, or you are running your containerized workload in a multi-cloud or hybrid cloud environment, as you will have better control over managing the microservices running across various platforms of CSPs. Even though Cloud Run is more serverless, it still provides some control on selecting the programming languages or binaries. Therefore, it has been kept under the CaaS category.

Function-as-a-Service (FaaS): You select the Function-as-a-Service option when you want to run a job for a specific time to complete some specific task. This is done mostly in response to some specific event. Cloud functions is a service available in Google Cloud Platform that is for this kind of requirements.

Platform-as-a-Service (PaaS): You opt for the Platform-as-a-Service offering when you do not want to manage the underlying platform and infrastructure. You just have an application to run and make use of PaaS offerings like an app

engine for running an application. App Engine is a preferred option for running a monolith application as it provides scalability with no management overhead.

So, we select the appropriate option depending on the use cases as described previously. Apart from this, selecting the appropriate compute option also depends upon the number of skilled resources and deployment timeline that you have for deploying an application. So development and operations teams need to discuss together before selecting any cloud-to-compute option.

Lab exercises

Since the purpose of this book is to familiarize the readers of Google Cloud Compute offerings, we will touch upon a few basic lab exercises. We will start with creating an account in GCP. Post that, we will see how to create a compute instance and a standard GKE cluster.

Creating an account in the Google Cloud Platform

In this lab exercise, we are going to perform the steps required to create an account in the Google cloud platform.

1. Login to Google Cloud Console **https://cloud.google.com/** and click on **`Get started for free`**. Refer to *figure 2.2*:

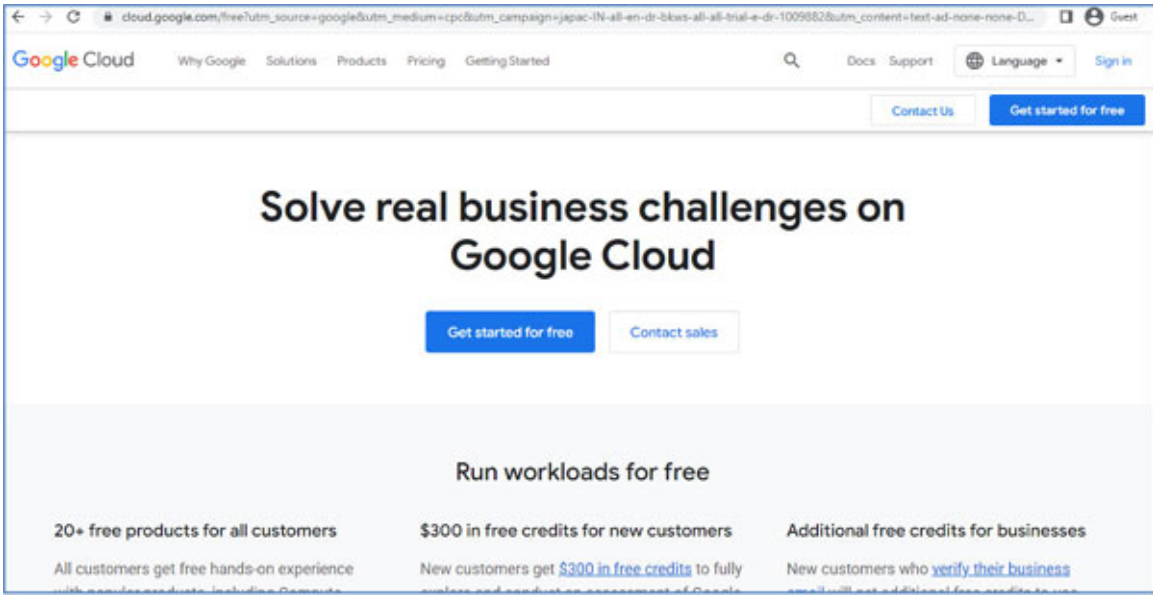

Figure 2.2: *Logging in to Google cloud console*

2. You will now be asked to login using your Gmail credentials, as shown in *figure 2.3*:

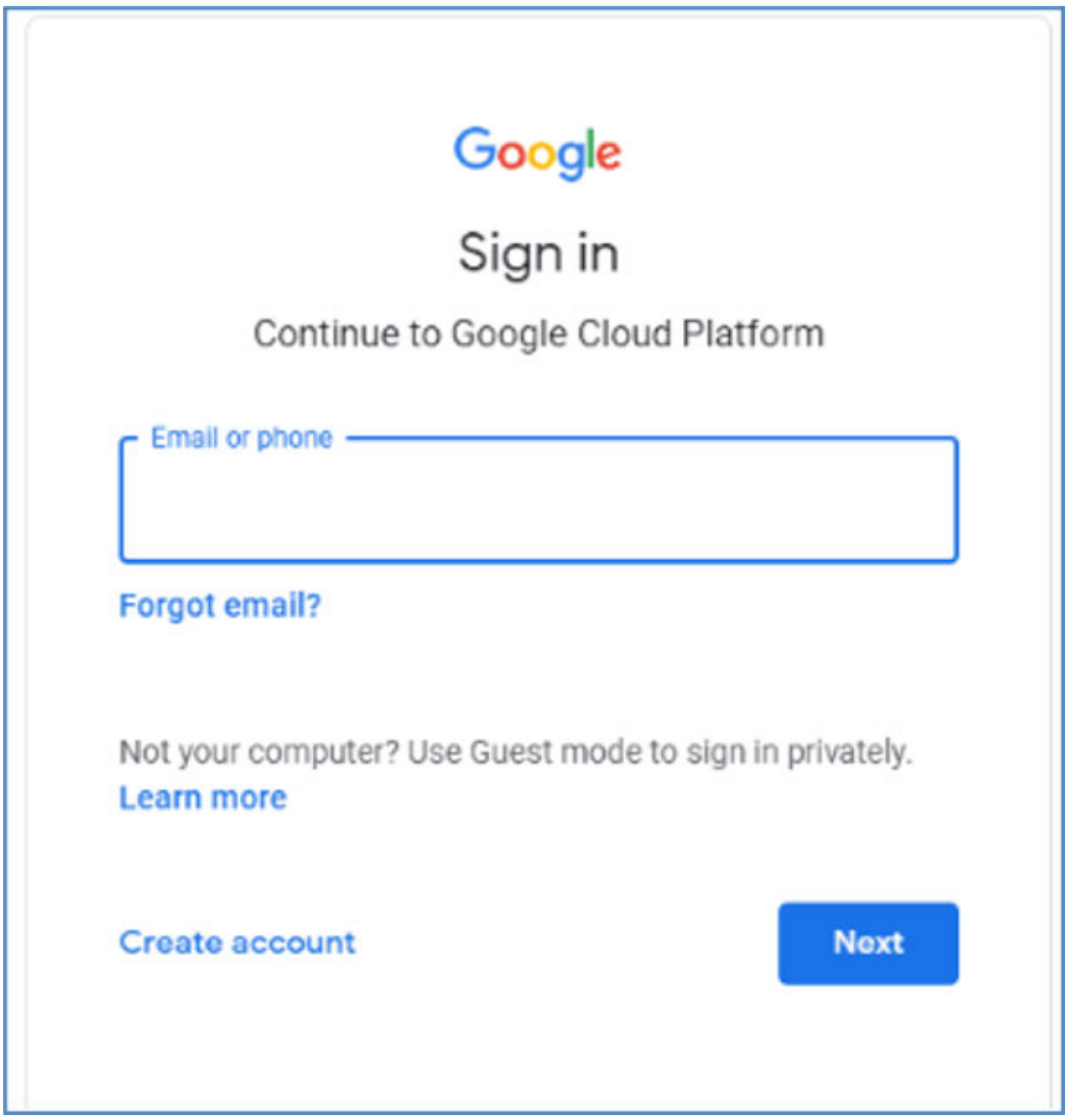

Figure 2.3: *Login using Gmail credentials*

3. On the next page, select your country and accept the Terms of Service. Click on **`Continue`**. Then you will get an option to fill in your details such as account type, Name, Address, credit card details, and so on. Refer to *figure 2.4*:

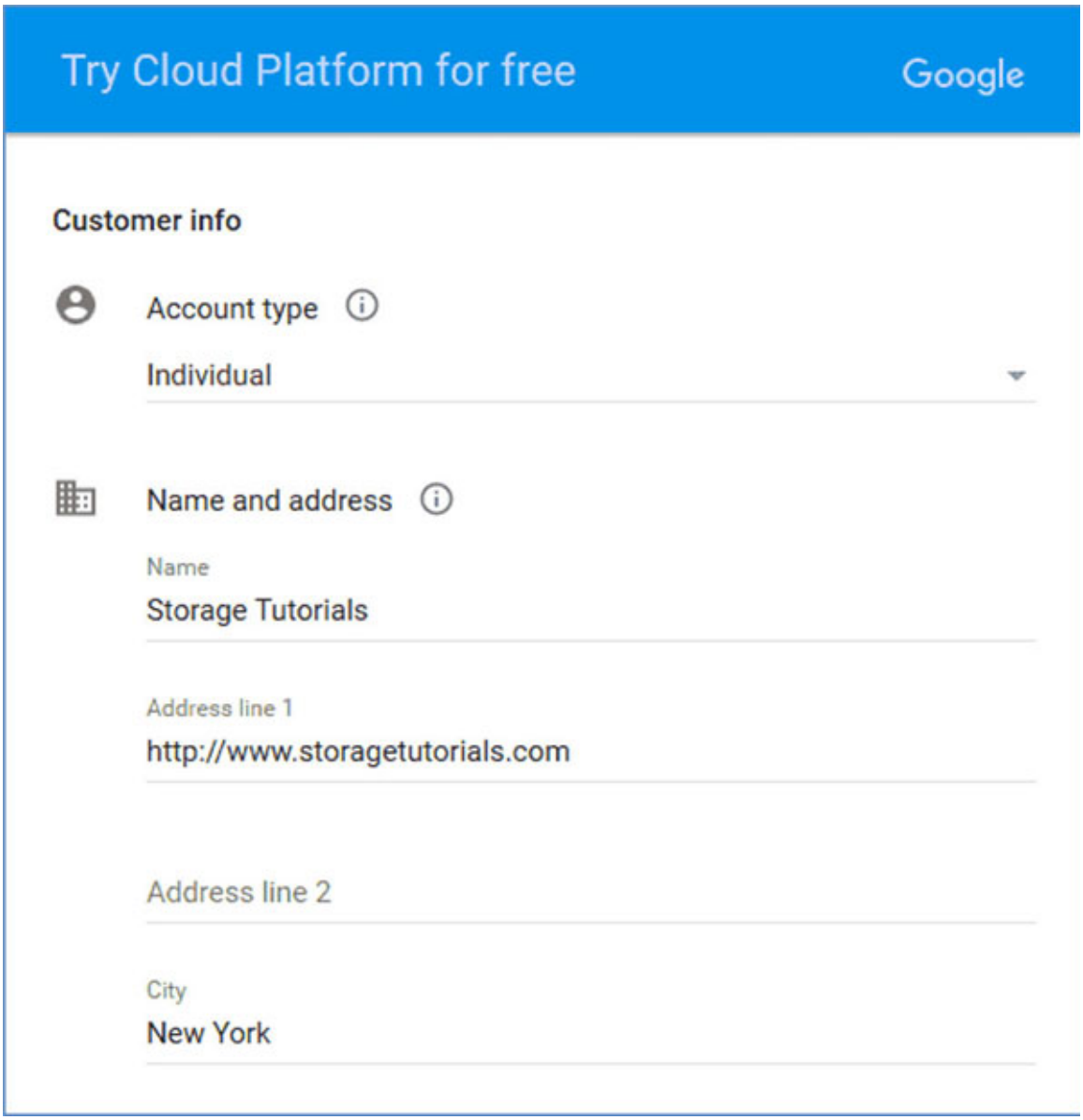

Figure 2.4: *Fill in the details*

4. After filling in your credit card details and click on **`Start my free trial`**, as shown in *figure 2.5*:

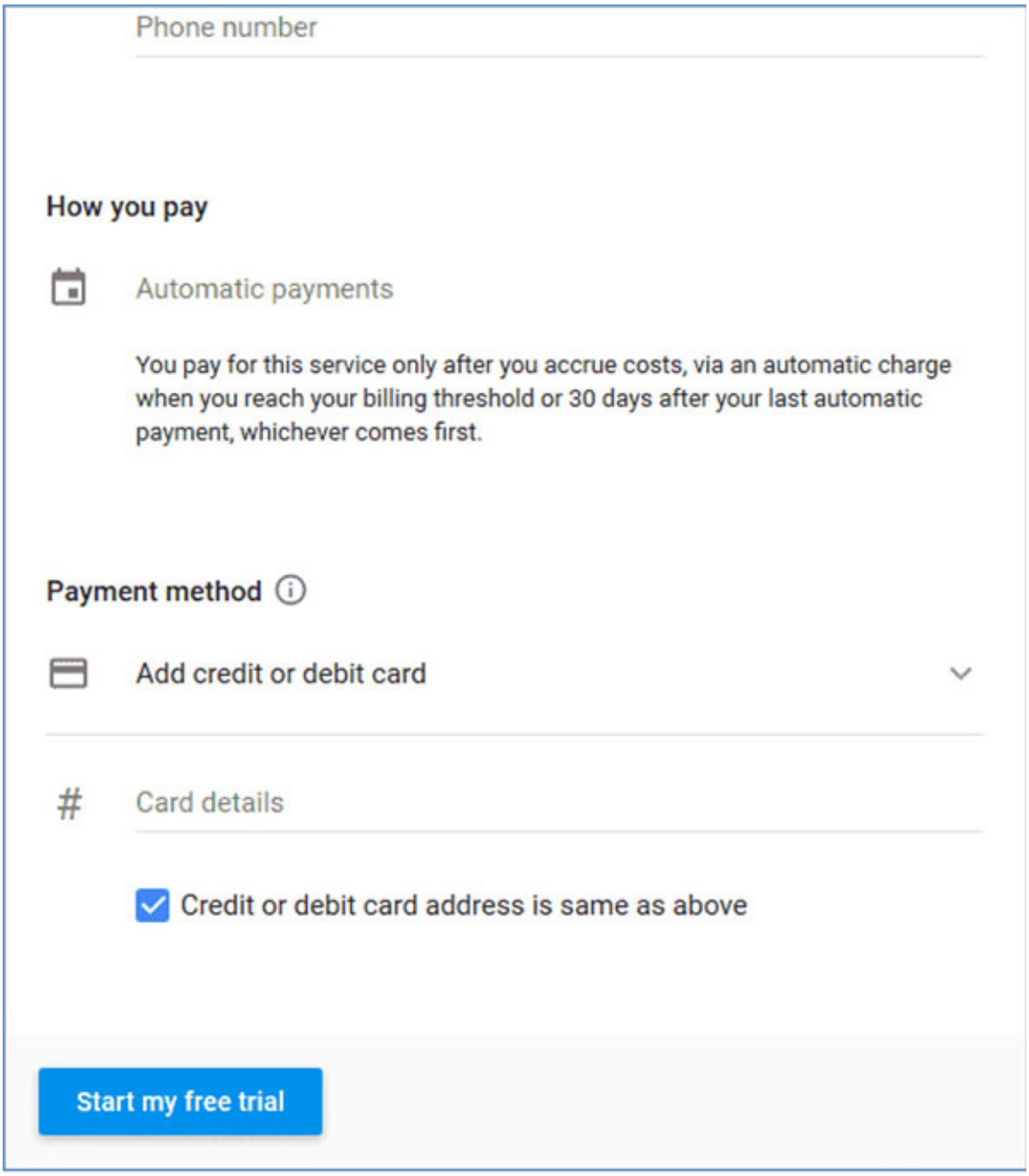

Figure 2.5: *Start your free trial*

5. Within a few seconds, your Google Cloud account will be setup, and the Google Cloud Platform will be ready to be used. Refer to *figure 2.6*:

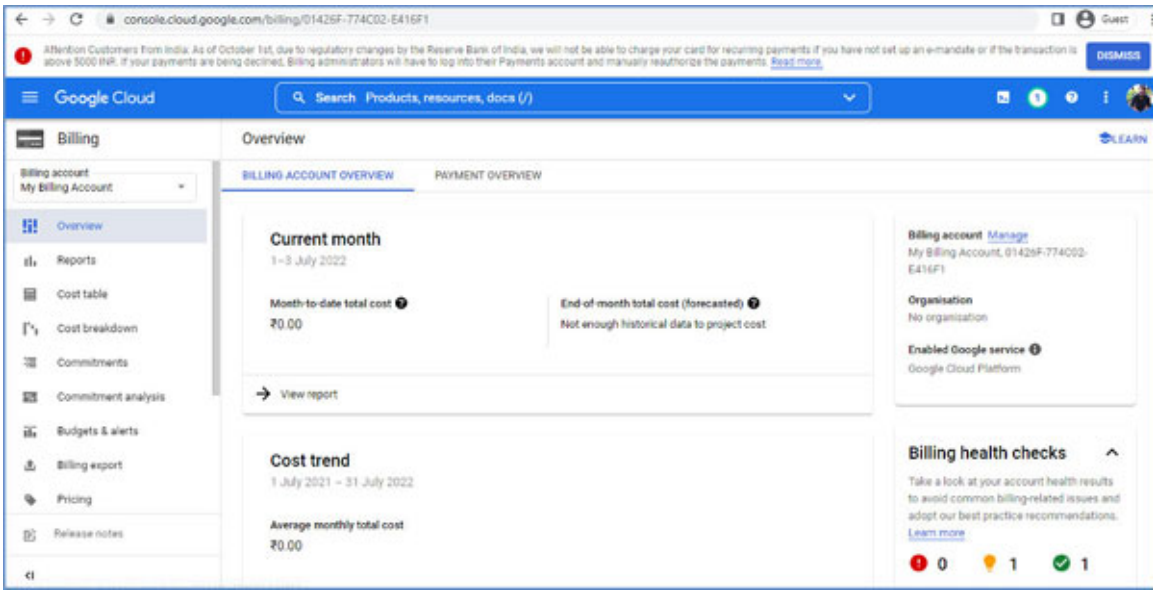

Figure 2.6: *Ready to use Google Cloud Platform*

Creating a Compute Instance

In this lab exercise, we are going to perform steps to create a compute instance in the Google cloud platform. Follow the following steps:

1. Go to the Navigation menu and select **`VM instances`** after clicking on **`Compute Engine`**. Since you are using it for the first time, you may be asked to enable "Compute Engine API". If prompted, click **`Enable`** to enable the API. Enabling APIs sometimes takes unto 5 minutes. Refer to *figure 2.7*:

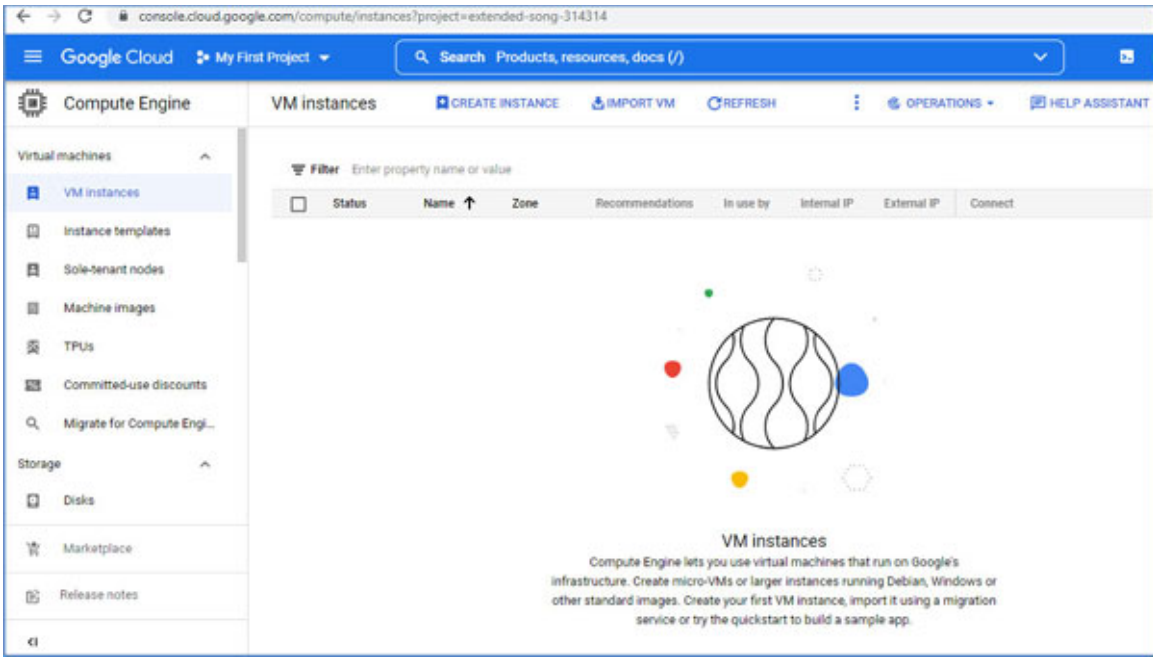

Figure 2.7: *VM Instances in GCP*

2. Type the instance name and select region and zone. Please be careful while selecting **"Regions"** and **"Zones"**, as these settings cannot be changed at a later stage. Refer to *figure 2.8*:

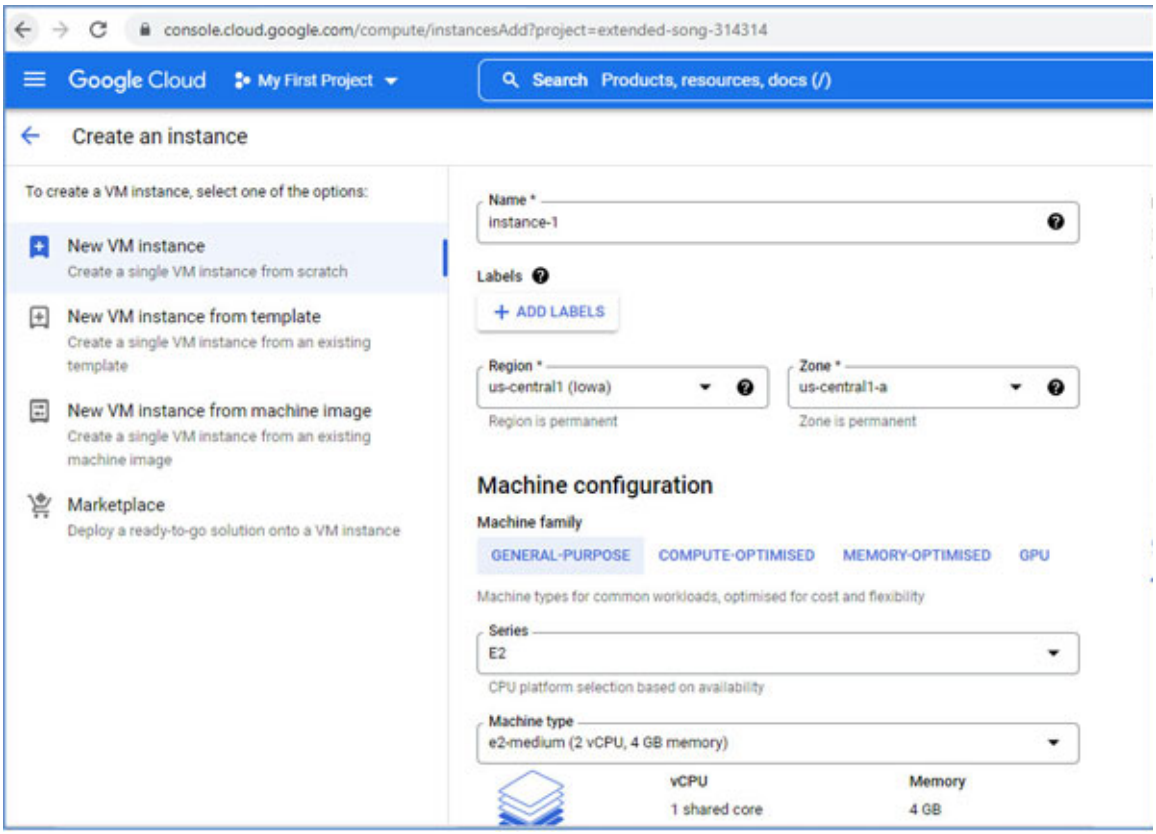

Figure 2.8: *Entering instance name, region, and zone*

3. In case you want to change the boot disk configuration, you can do so by clicking on **`Change`** Select **`Boot Disk`** configuration after clicking on **`Change`**, as shown in *figure 2.9*:

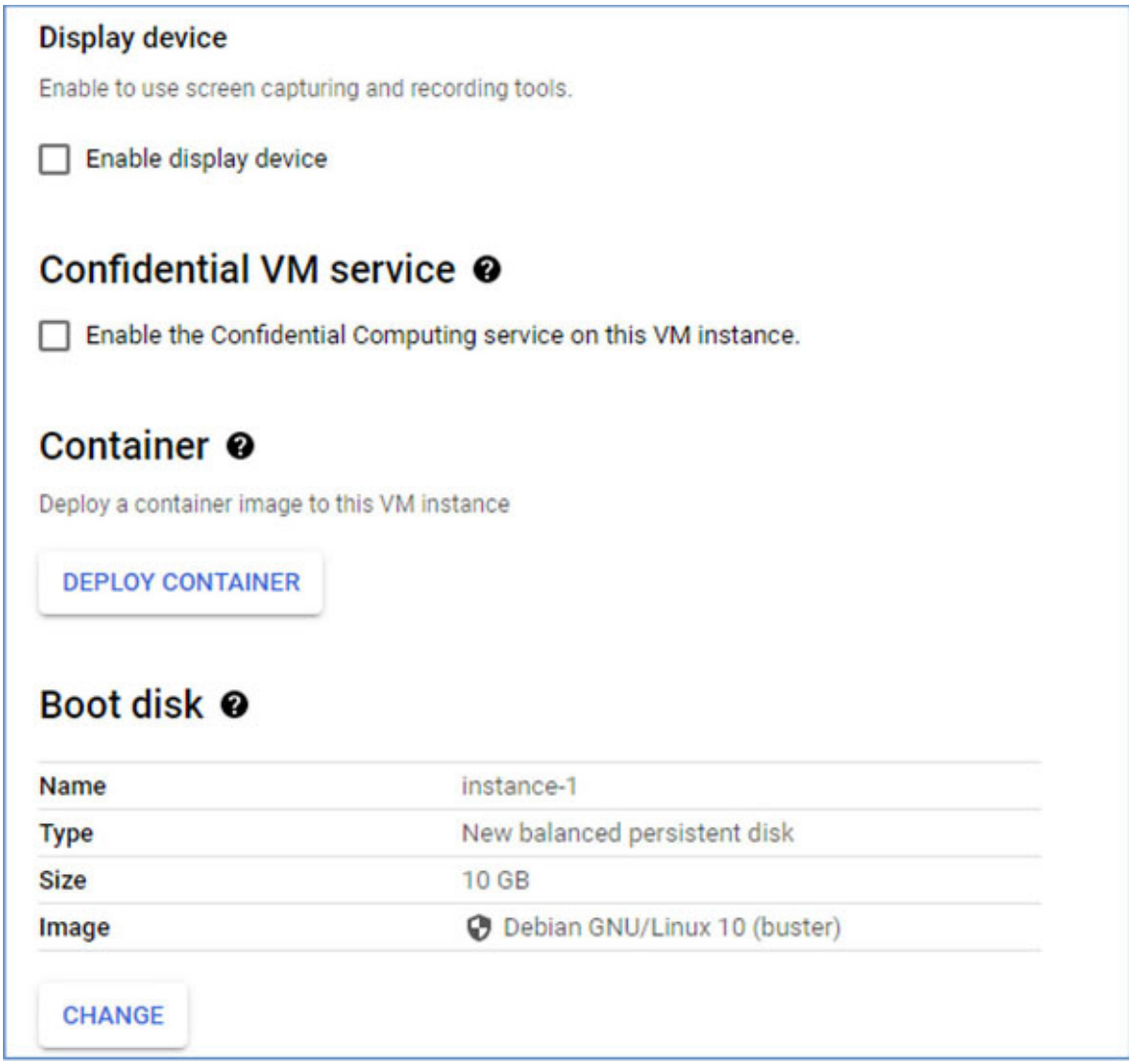

Figure 2.9: *Changing book disk configuration*

4. After clicking on **`Change`**, you will get options to select Operating System, Version, disk type, and so on. Make the changes and click on **`Select`**, as shown in *figure 2.10*:

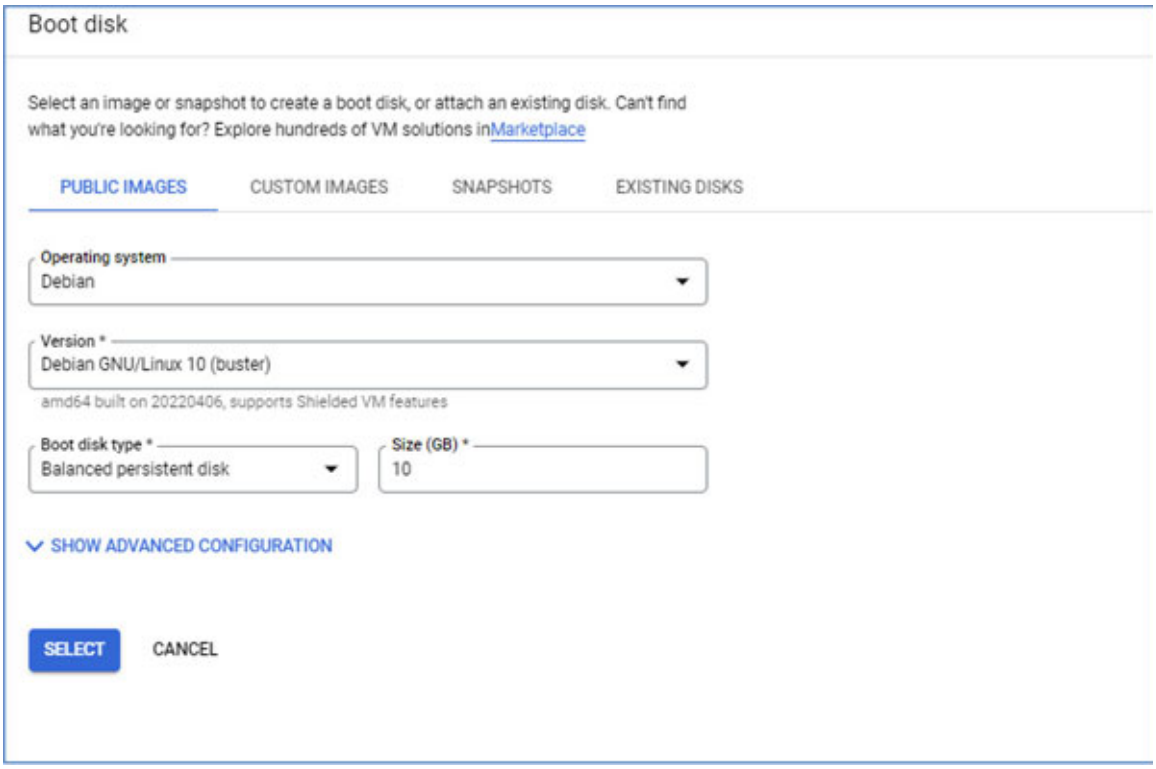

Figure 2.10: *Enter details of the operating system, version, boot disk type, and size*

5. Select the appropriate "API access" from the available scope. This is primarily used for development purposes. Also, select the Firewall option depending upon whether you need http or https access, and click on **`Create`**. Your virtual machine will get created. Refer to *figure 2.11*:

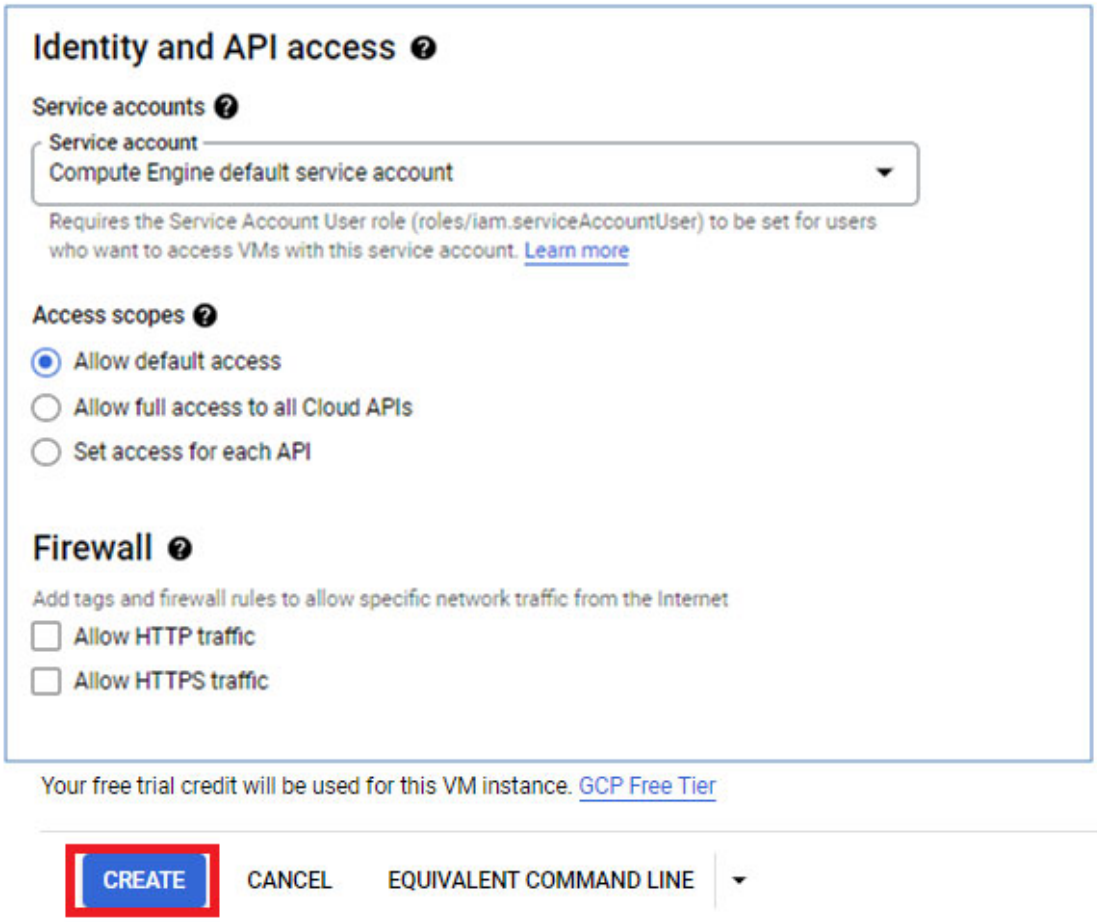

Figure 2.11: *Creating the virtual machine*

Creating a Standard Google Kubernetes Engine (GKE)

Here, we are going to perform the following steps to create a standard GKE cluster.

1. Go to the navigation menu and select **`Kubernetes Engine`**|**`Clusters`**, as shown in *figure 2.12*:

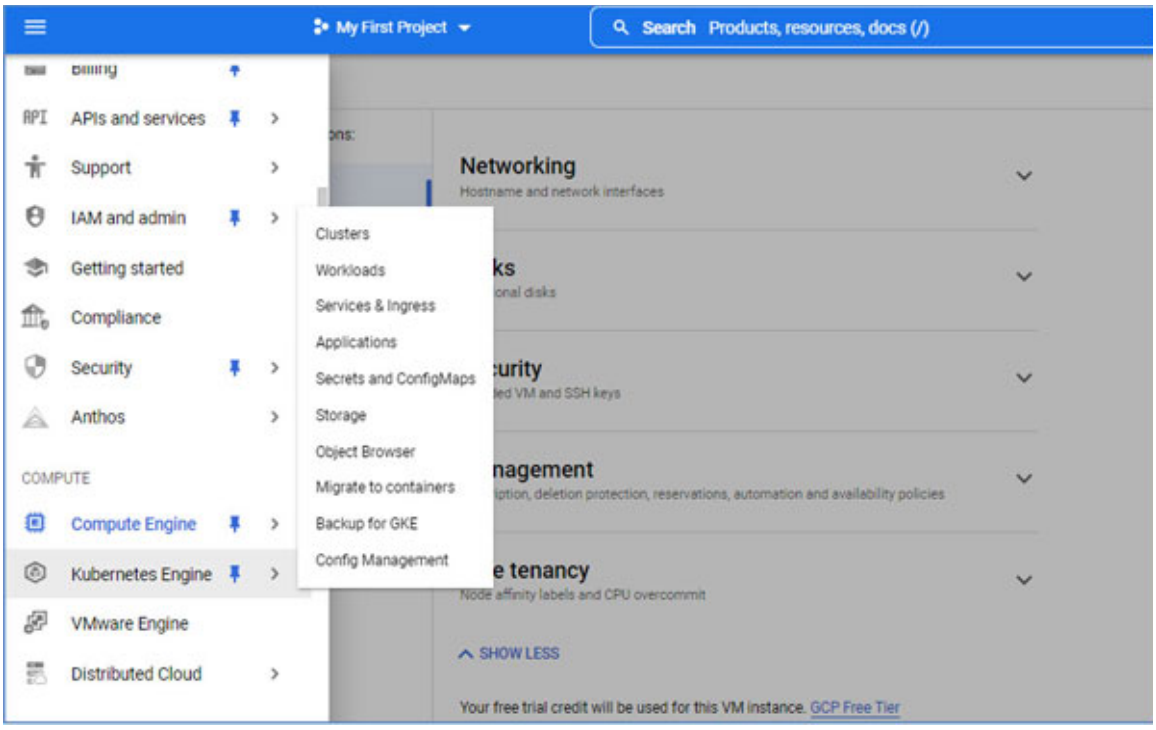

Figure 2.12: *Creating a standard GKE*

2. In the next screen, click on **`Create`**, as shown in *figure 2.13*:

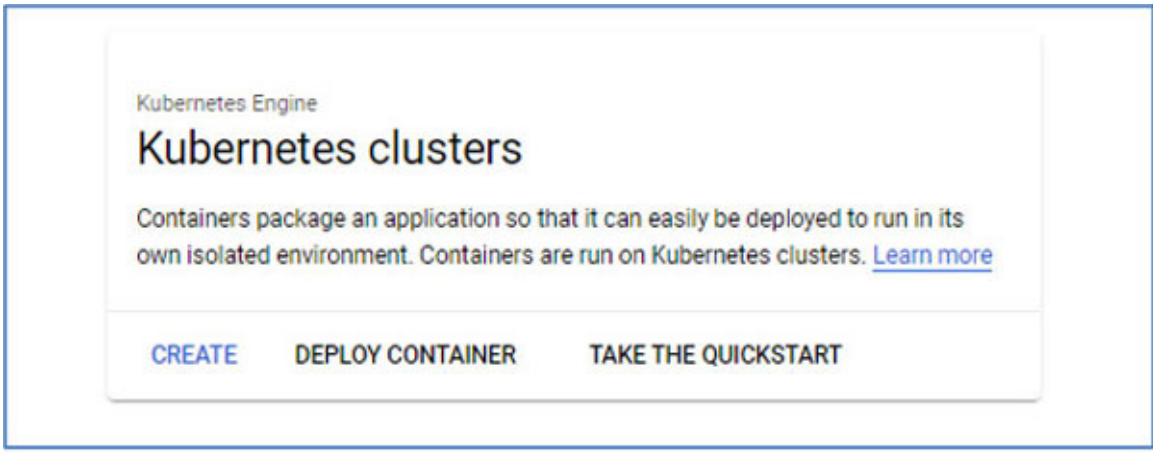

Figure 2.13: *Click on create*

3. Now, click on **`Configure`** against **`GKE Standard`**, as shown in *figure 2.14*:

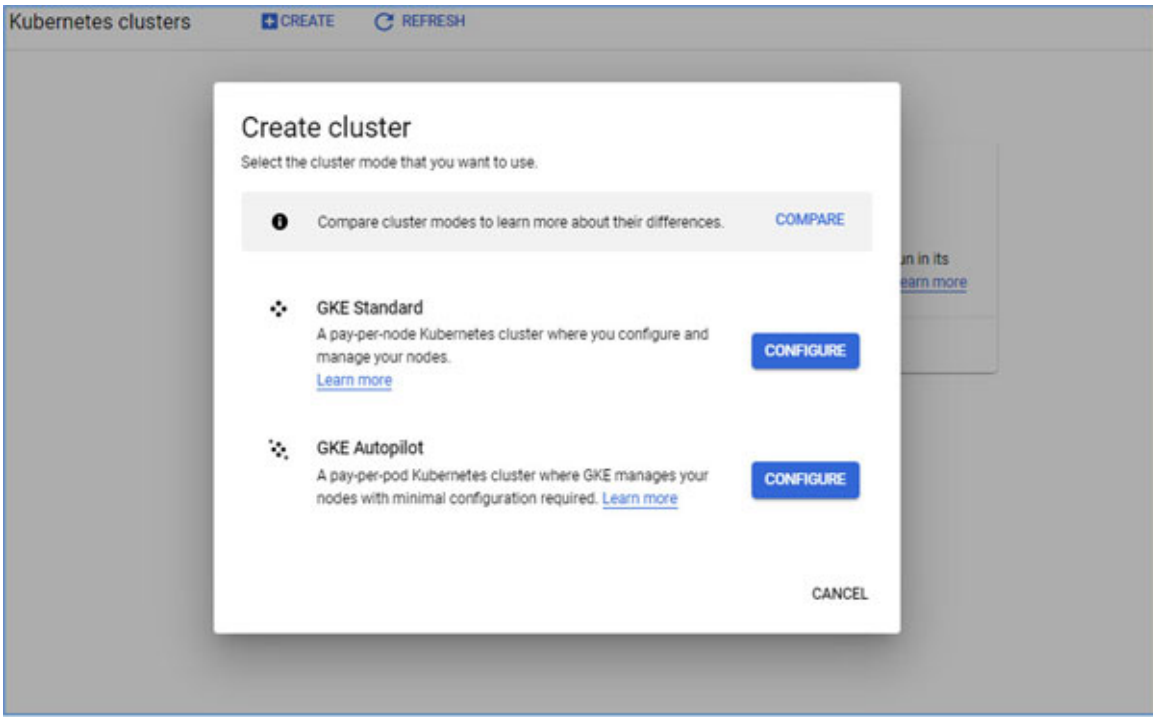

Figure 2.14: *Click on configure to be able to configure nodes in a pay-per-node Kubernetes cluster*

4. In the next screen, please select the **`Name`** of the cluster, **`Location type`**, and the control plane **`Version`**, as shown in *figure 2.15*:

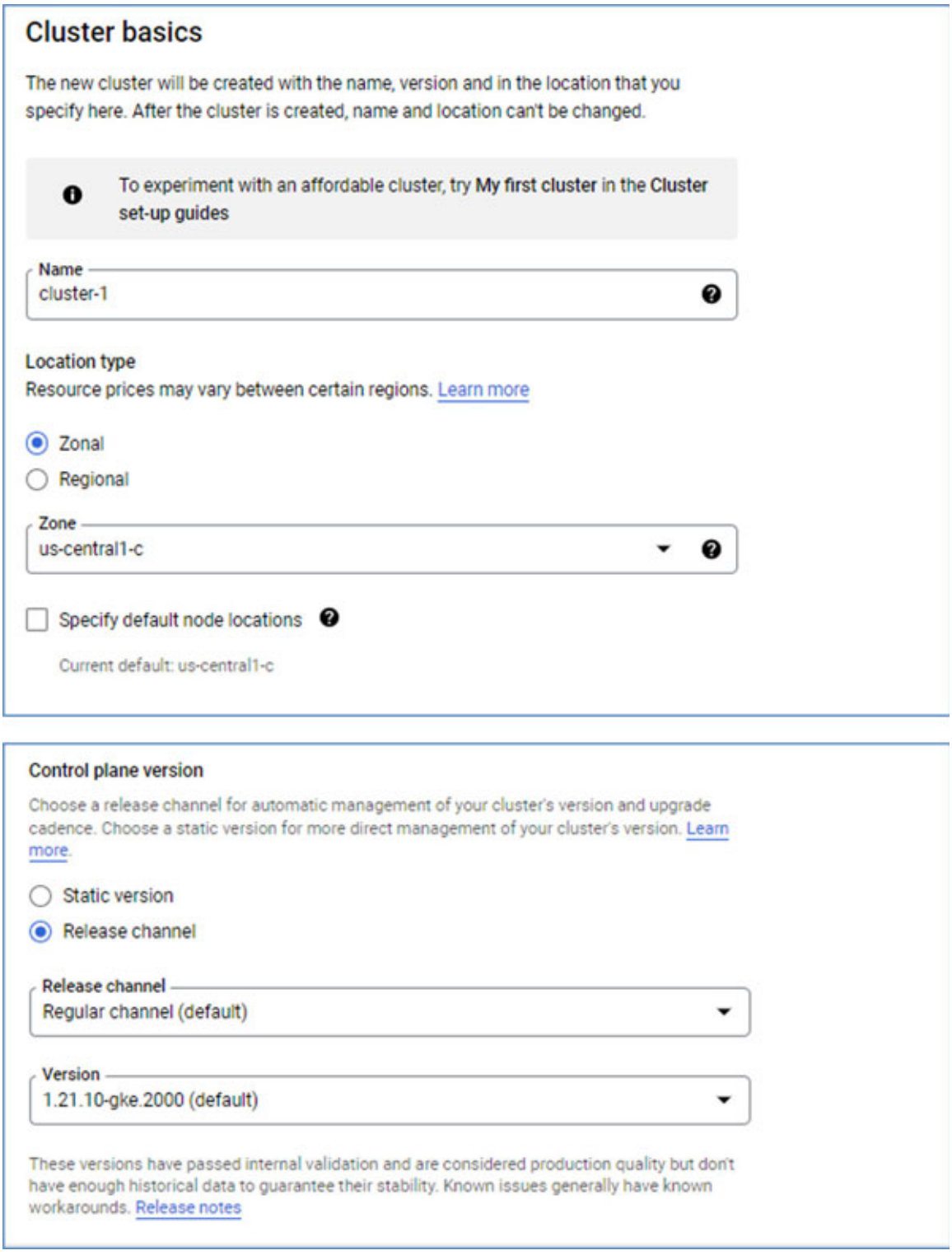

Figure 2.15: *Add details of the cluster*

5. Now click on **`Default-pool`** option under **"Navigation Menu".** Here, you get an option to select a number of **"Nodes"** in a **"Node Pool"**, and

you get an option to enable **"Autoscaling"**. You can also specify **"Node Locations"** and select one or more locations where you need to run your workload. You need to have at least one **"Node Pool"** inside your Kubernetes Cluster.

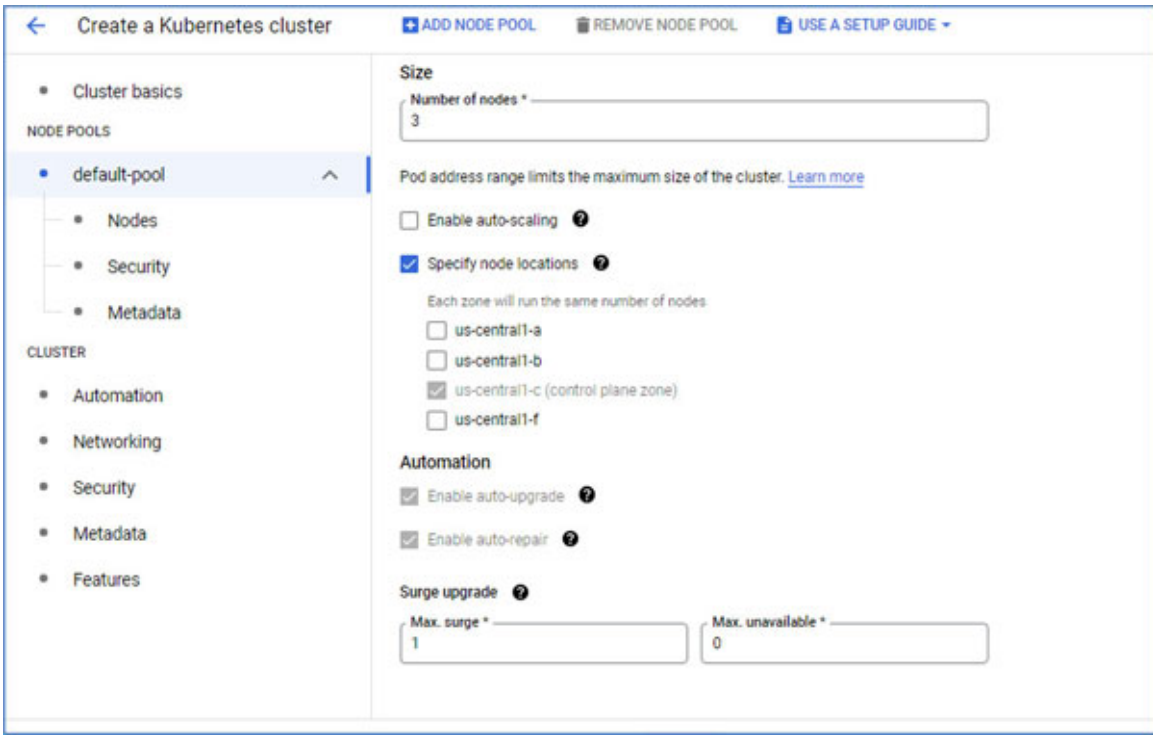

Figure 2.16: *Add details for the node pool*

6. Now click on **`Nodes`**, and you get an option to select the **`Machine Type`**. You can also select a **`Boot disk`** for your requirement, as shown in *figure 2.17*:

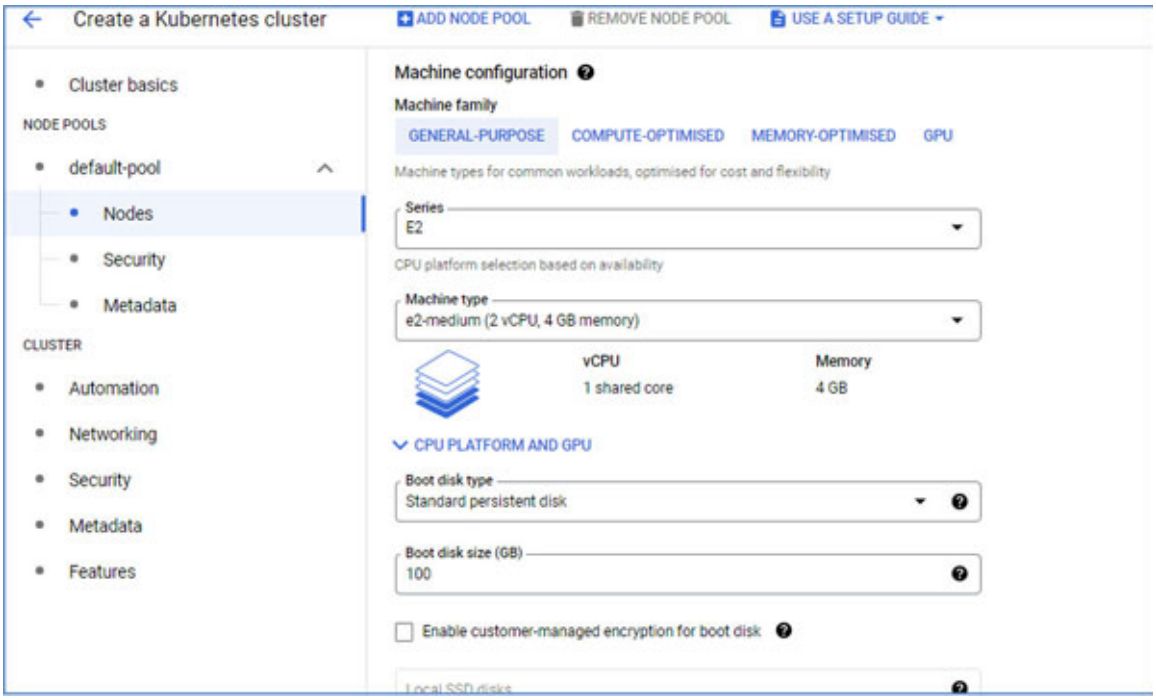

Figure 2.17: *Selecting the machine type*

7. Now click on **`Automation`** in the **"Navigation Menu"**. You get an option to select the **"Maintenance Window"** for your Kubernetes Cluster. You also get options like **"Enabling monitoring"** and **"Vertical autoscaling"** for your nodes and PODs.
8. Now click on the **`Networking`** option. You can either select a **"Default"** VPC network or any **"Custom"** VPC network that you might have created already. You also get an option to select it as a **"Public"** Cluster or a **"Private"** Cluster". Enable **"Enable VPC-native traffic routing (uses alias IP)"**. Click on **`CREATE`**. Refer to *figure 2.18*:

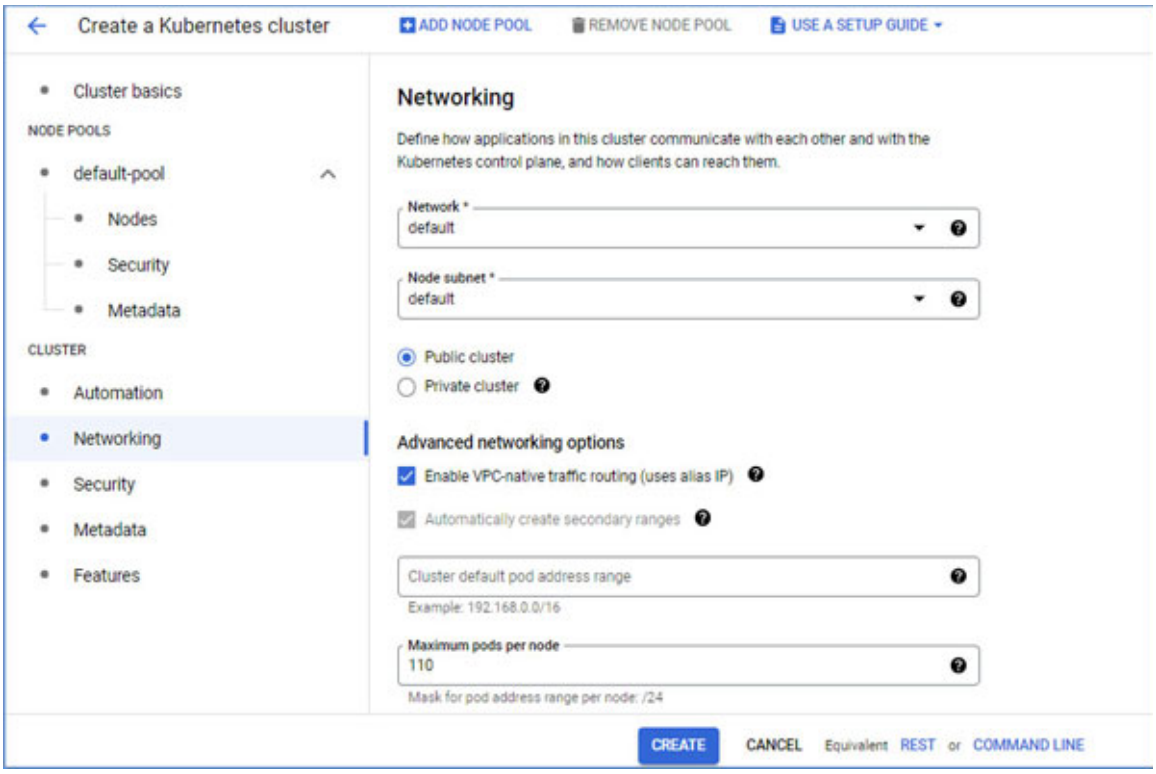

Figure 2.18: *Creating a standard GKE cluster*

9. In around 5 minutes, your standard GKE cluster will be created.

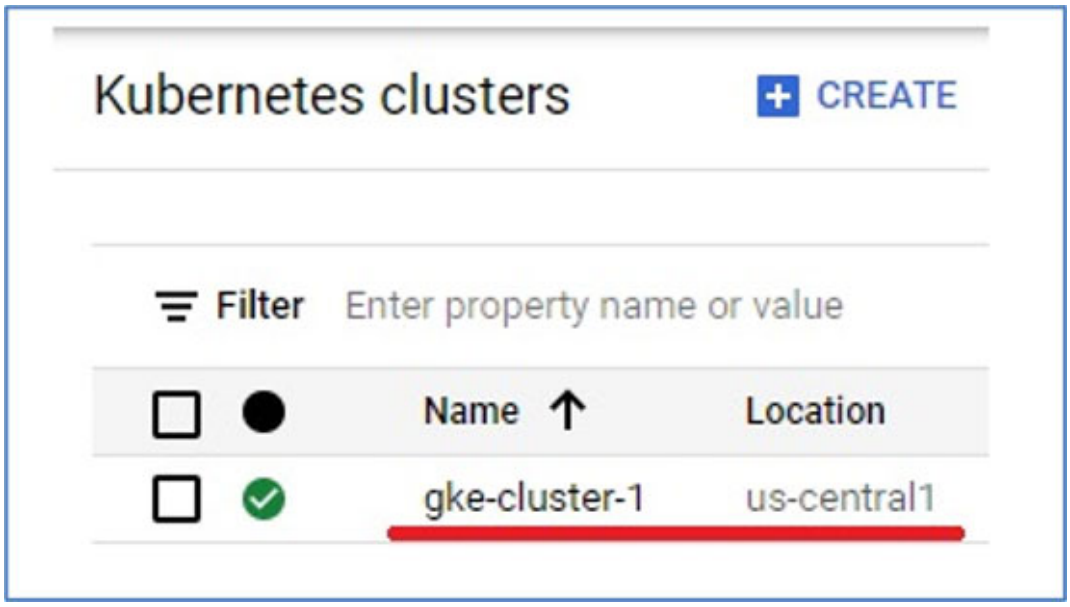

Figure 2.19: *GKE cluster is created*

Conclusion

As mentioned at the beginning of this chapter, compute is one of the most important resources in any type of data center. Selecting the right compute resource is very much essential in a cloud environment as using the right resource, the benefits of cloud computing can be leveraged appropriately. Selecting the wrong compute can impact the functioning of the workload fora longer duration in many different ways, such as performance, pricing and so on. Compute is a broad topic and requires further reading and practice. Therefore, readers are advised to go through Google Cloud documentation and have some hands-on practice in the lab environment. In the upcoming chapter, we will discuss about various storage options available in Google cloud. Readers will also be able to understand the various use cases related to various storage services available in the Google Cloud platform.

Key terms

- **FaaS**: Function-as-a-Service.
- **CaaS**: Container-as-a-Service.
- **GKE**: Google Kubernetes Engine.
- **Docker**: Open source containerization platform used to package application code with OS libraries and dependencies.
- **POD**: A collection of one or more containers. Smallest deployment running inside the nodes of Kubernetes cluster. It contains one or more containers.
- **Stateless applications**: An application that does not require data generated from one client session to be utilized in another session.
- **Stateful applications**: An application that requires data generated from one client session to be utilized in the subsequent session.

Questions

1. Mention various compute options available in the Google Cloud Platform.
2. What is a Compute Engine, and what are its various families and types?
3. Explain the architecture of a Kubernetes Engine and why it is used.
4. What is GKE, and what are its types?

5. What is Cloud Run, and how it is different from Google Kubernetes Engine?
6. Explain App Engine and its use.
7. Which compute option can be used as a Function-as-a-Service (FaaS) offering?

Further reading

- **https://cloud.google.com/blog/products/compute/choosing-the-right-compute-option-in-gcp-a-decision-tree**

CHAPTER 3

Storage in Google Cloud

Cloud storage is the abstraction, pooling, and sharing of storage resources through the internet. Cloud storage is facilitated by IT environments known as clouds, which enable cloud computing—the act of running workloads within a cloud environment.

RedHat

Introduction

Storage technology has gone through many significant changes over the last few decades. From traditional SATA drives for internal storage and floppy drives for external storage to the SSD drives and flash cards. With technological advancement and reduced pricing, we have seen a significant increase in the usage of SSD drives. Now with the cloud, we are seeing various cloud providers providing storage as a service wherein you have the option to store your data in the cloud. In this chapter, we will understand the various types of storage services available in the Google Cloud Storage.

Structure

In this chapter, we will cover the following topics:

- Storage and its types
- Various Storage options in Google Cloud
- Storage encryption in Google Cloud
- Choosing the right storage option
- Lab Exercise

Objectives

After reading this chapter, readers will be able to get a fair idea about various storage options available in the Google Cloud Platform. Readers will

understand when to use which storage service for their requirements. There is a small lab exercise available for the readers that they can use to get some hands-on in the Google Cloud Platform.

Storage and its types

We have been using locally attached storage in the form of a disk drive. Starting from storing operating system images, applications, and other data in the form of files and directories, everything used to get stored locally.

As the generation of data continues to grow exponentially, the requirement of data storage devices that can store a large amount of data with faster access capability has increased. This has resulted in innovation in this area. Also, as analytics is playing a major part in any of businesses, the need for more advanced data storage devices has increased. With cloud technology coming into the picture, various cloud service providers have started providing storage as a service, where customers pay for storage based on usage, otherwise called the pay-as-you-go model.

In general, storage can be categorized into the following three categories:

- **Block storage:** In Block storage, we have to store the data in the form of data blocks. Your disk drive that is connected to your desktop or laptop directly is an example of block storage. We also have centralized storage, which is also called a **Storage Area Network** (**SAN**), wherein data is stored at a centralized location and can be accessed by many servers. In Block storage data is read and written in the form of data blocks.
- **File storage:** When you want to share files with multiple servers, you make use of File storage. You have a shared file system that is accessed by multiple servers. In File storage, data is stored in a hierarchical structure in the form of files and directories. **Network Attached Storage** (**NAS**) makes use of file storage. File storage also provides you with an option of access control. We make use of NFS and CIFS protocols in order to share the files across operating systems.
- **Object storage**: When you want to store a large amount of unstructured data, Object storage is the preferred choice. Object storage stores the data in the form of objects, and there is an identifier associated with it. Some examples of unstructured data that should be stored in Object storage are video files and documents. Data in object storage is not stored in the form of files and directories but resides in the storage buckets. All information

about the data is stored in the form of metadata. When you want to store unstructured data that has a low latency requirement, then you should prefer Object storage. Object storage is also used for backup and archival.

Various storage options in Google Cloud

GCP provides us with the following storage options. These storage options are chosen for specific requirements:

- **Local SSD:** Local SSD is the storage that is directly attached to the physical server on which virtual machines are hosted. Local SSD provides extremely low latency and high IOPS. Due to this, it is suitable for high-performance computing or analytics, and so on. Local SSD is temporary or ephemeral in nature, which means if you delete a VM instance, local SSD also gets deleted. Maximum size of a local SSD is 375 GB; however, the total storage size supported by local SSD is 9 TB per instance. You can add up to 24 SSD partitions per VM instance.
- **Persistent disk**: Persistent storage can be considered as an example of SAN storage in which the data is stored in the form of blocks, but the storage disks are not directly attached to your virtual machines. The size of the disk and compute of the virtual machine increases, and the performance of the persistent disk also Increases. You can create snapshots of your persistent disks and restore them at a later stage. You can install your applications and databases on your persistent disks. Even if your virtual machines get deleted, you still get an option to persist your persistent disk. As of date, Storage size supported by the Persistent disk in GCP is 64 TB.
- **Cloud Storage**: When you are looking for an Object storage option in the Google Cloud Platform, you opt for Cloud Storage. Unlike block and file storage options, where the data is stored in the form of files and directories, Cloud Storage stores data in the form of objects. Here, data is not stored in a hierarchical structure; instead, it is stored in key-value format. This results in high read performance. Cloud Storage is used for storing unstructured data or for backup and archival purposes. Some examples of Cloud Storage are video files or documents in CSV format. Cloud Storage stores data inside buckets. Cloud Storage provides high durability, availability, and unlimited storage. There is no limitation on the amount of data that can be stored in Cloud Storage. We have various

storage classes available in Cloud Storage that can be used for a specific purpose and provides a nice combination of performance and cost:

- **Standard storage class**: This storage class is used for storing frequently accessed data. Online video streaming platform stores videos or movies, and static websites are some examples of the data that is stored in standard storage class. Standard storage class can be of three types, regional, dual-region, and multi-region, offering 99.9%, 99.95%, and 99.95% availability SLAs, respectively.
- **Nearline storage class**: Nearline storage is another storage class that is used for infrequently accessed data. When your requirement is to access the data once in a month or less, the nearline storage class is a better choice as it is relatively cheaper to store the data in this storage class. Thirty days is the minimum storage duration in this storage class. Nearline storage classes also come with regional, dual-region, and multi-region options offering 99.0%, 99.9%, and 99.9% availability SLAs, respectively.
- **Coldline Storage class**: Coldline storage class is used to store the data that needs to be accessed once in a quarter, that is, 90 days. Coldline storage is cheaper than Standard and Nearline storage classes. Ninety days is the minimum storage duration in the coldline storage class. When you plan to read or change the data once in a quarter, you opt for this storage class. Under regional, dual-region, and multi-region options, coldline storage class provides 99.0%, 99.9%, and 99.9% availability SLAs, respectively. It can be used for backup and archival purposes.
- **Archival storage class**: As the name suggests, Archival storage is used for data Archival and when your data access requirements are more than once in a year (that is, 365 days). Archival storage is the cheapest of all storage classes. The minimum retention period for storing the data in Archival storage is 365 days. Archival storage can also be used for Disaster Recovery. Archival storage also comes with regional, dual-region, and multi-region options providing 99.9%, 99.9%, and 99.0% availability SLAs, respectively.

Note: Even though storage cost reduces according to the data retention and accessibility requirement, data access cost increases. So read/write activity in an Archival storage class will cost you more than in a standard storage class.

- **Filestore**: When we want to share files or directories across multiple servers or applications, filestore is the correct option. It is a managed service that is used to share the file system across multiple applications or servers. This service gives you an experience of **Network Attached Storage** (**NAS**) that has been used traditionally to share a file system across multiple applications and servers. This service provides us with low latency.

Storage encryption in Google Cloud

All data in the Google Cloud Platform is encrypted by default. Google cloud uses the **Advances encryption standard** (**AES**) for encrypting data at rest. Google cloud platform has a service called Centralized Key Management Service where all encryption keys are Stored. In Google Cloud, data is written in the form of chunks. In Google Cloud, data is encrypted before it is written on these chunks and decrypted before it is read. Data written on each chunk is encrypted with a different encryption key. Whenever data in a chunk is updated, it is updated using a different encryption key. Google provides two options for encryption, one using Google-provided keys and another one with customer-provided keys. Apart from it, the customer also gets an option to manage these keys by themselves in case if they do not want Google to manage them.

Choosing the right storage option

Choosing the right option solely depends on your requirement or the use cases in hand. *Table 3.1* shows various use cases for a different type of storage options:

Categories	Feature	Use cases
Local Storage	Directly attached storage, High-performance storage	For temporary storage such as caching, local processing capacity, or storing the data that can be lost
Persistent Disk	Persistent and reliable storage	For storing applications and data
Cloud Storage	Secure, scalable storage for storing unstructured data	Streaming video and audio documents, storing data for backup, and archiving
Filestore	Fileshare access to the applications using the NFS protocol	For applications that use file shares, logs, and so on.

Table 3.1: *Cloud storage comparison*

Lab exercise

Let us now work on a few lab exercises.

Creating a Google Cloud Storage Bucket

To create a Google cloud storage bucket, follow the following steps:

1. Login to the GCP console. Click on the navigation menu and then on **Cloud Storage**, as shown in *figure 3.1*:

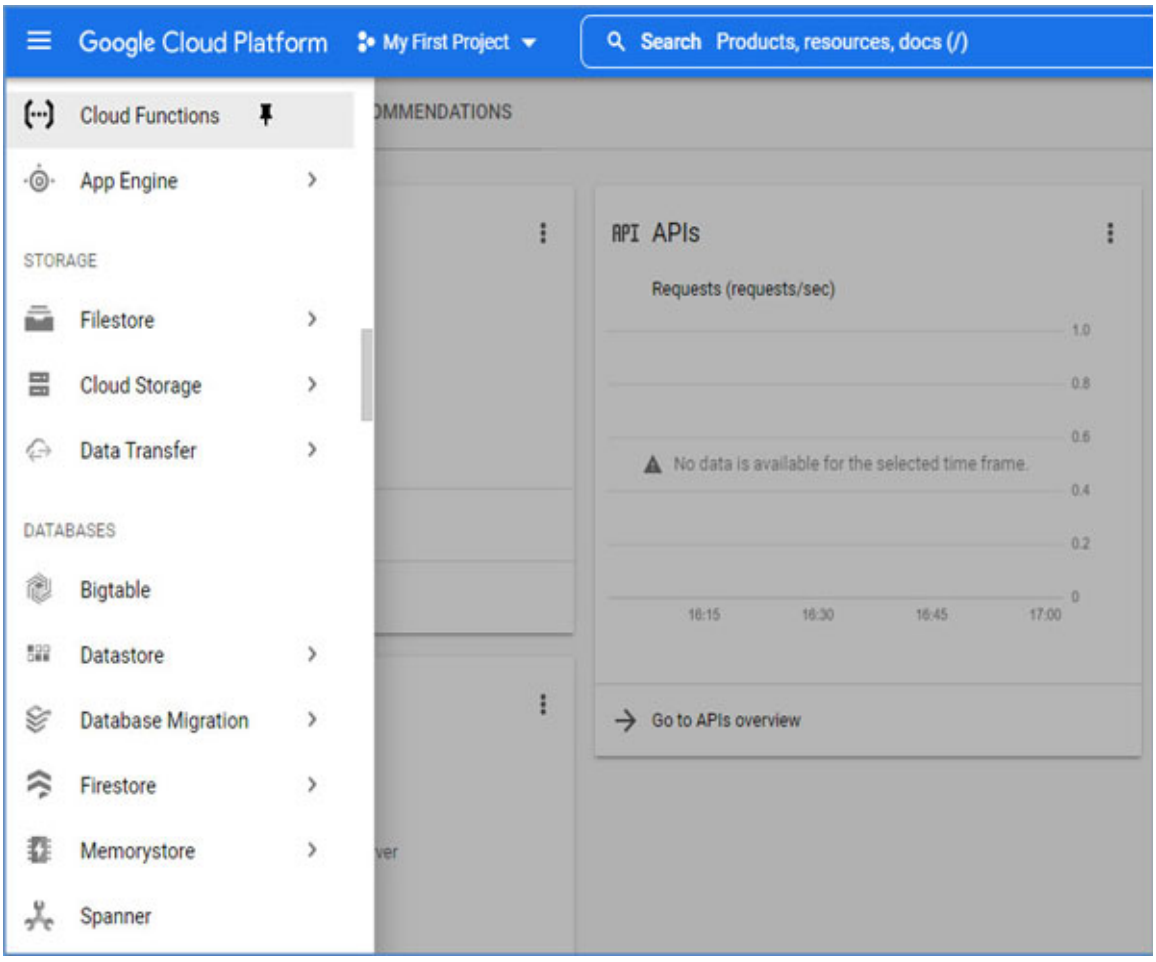

Figure 3.1: *Going to cloud storage in GCP console*

2. Click on **CREATE BUCKET**, as shown in *figure 3.2*:

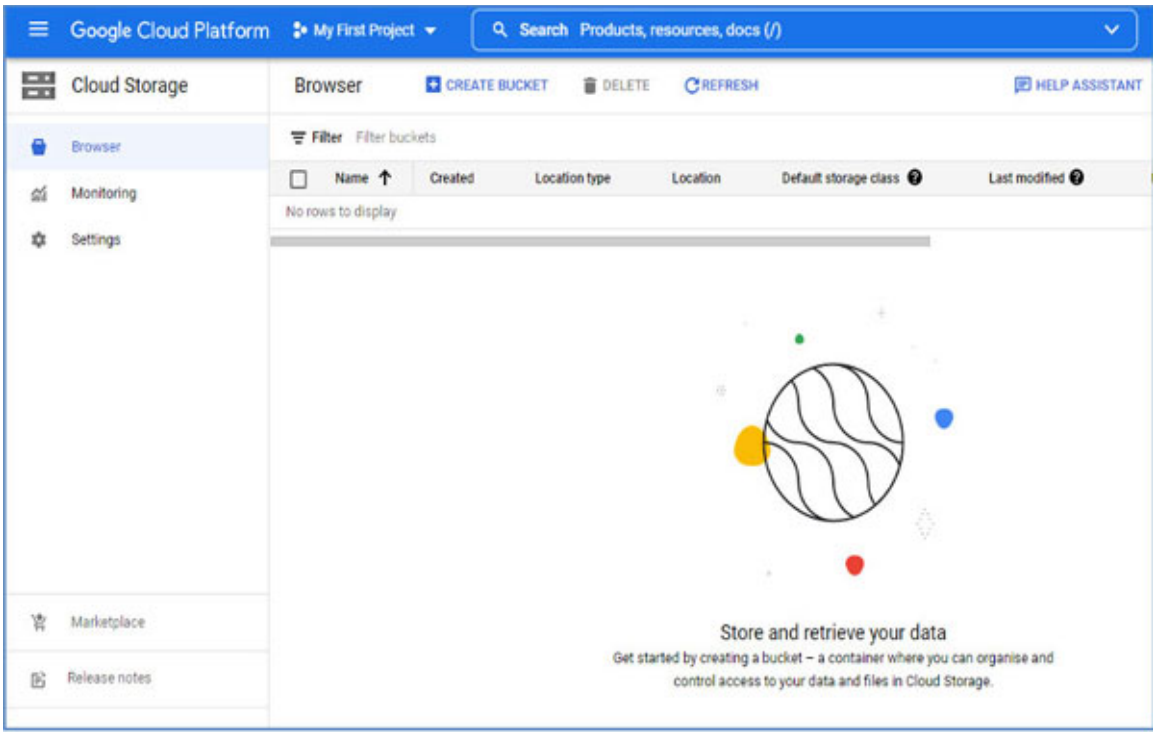

Figure 3.2: *Select "Create bucket"*

3. Give a unique name to the bucket and click on **CONTINUE**. Select **Regional** as location. Select any Region and click on **CONTINUE**. Refer to *figure 3.3*:

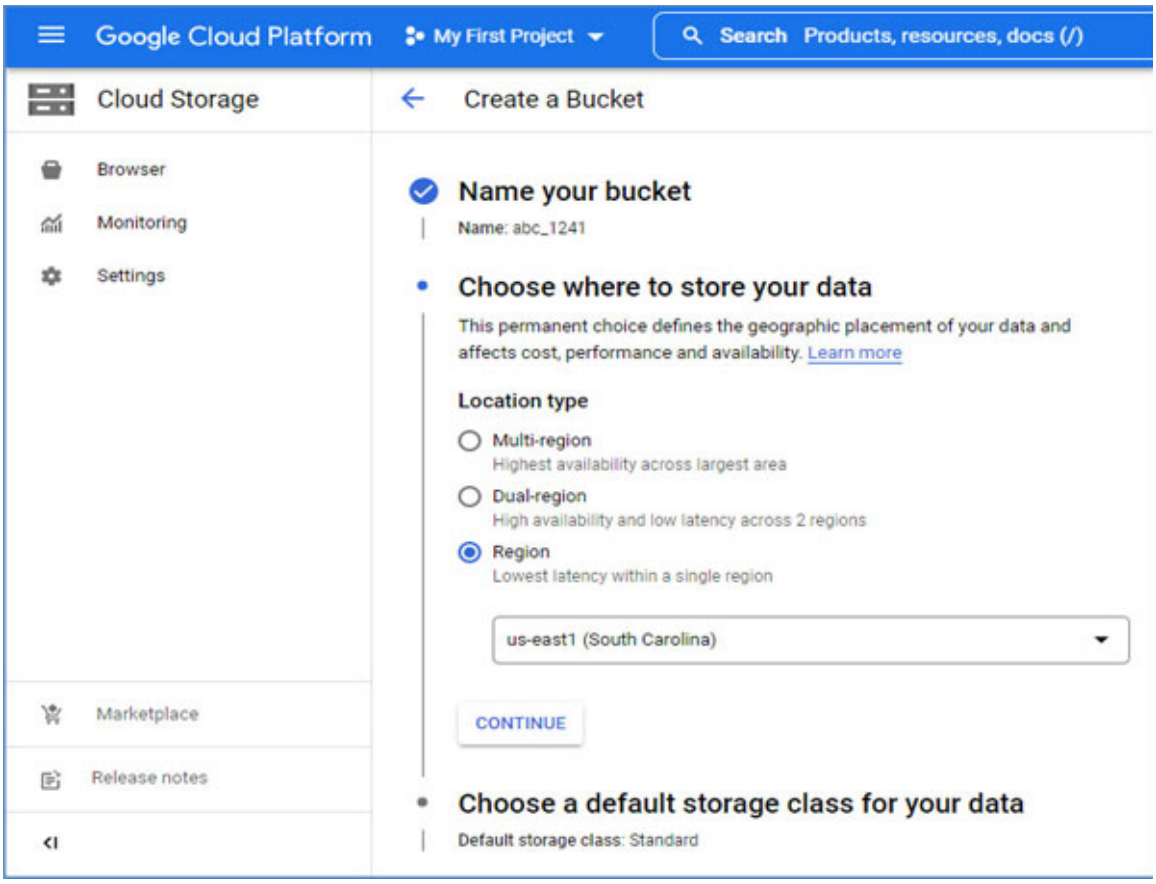

Figure 3.3: *Selecting a unique name and region*

4. Choose a default storage class. Select **Standard** as a default storage class and click on **CONTINUE**. Refer to *figure 3.4*:

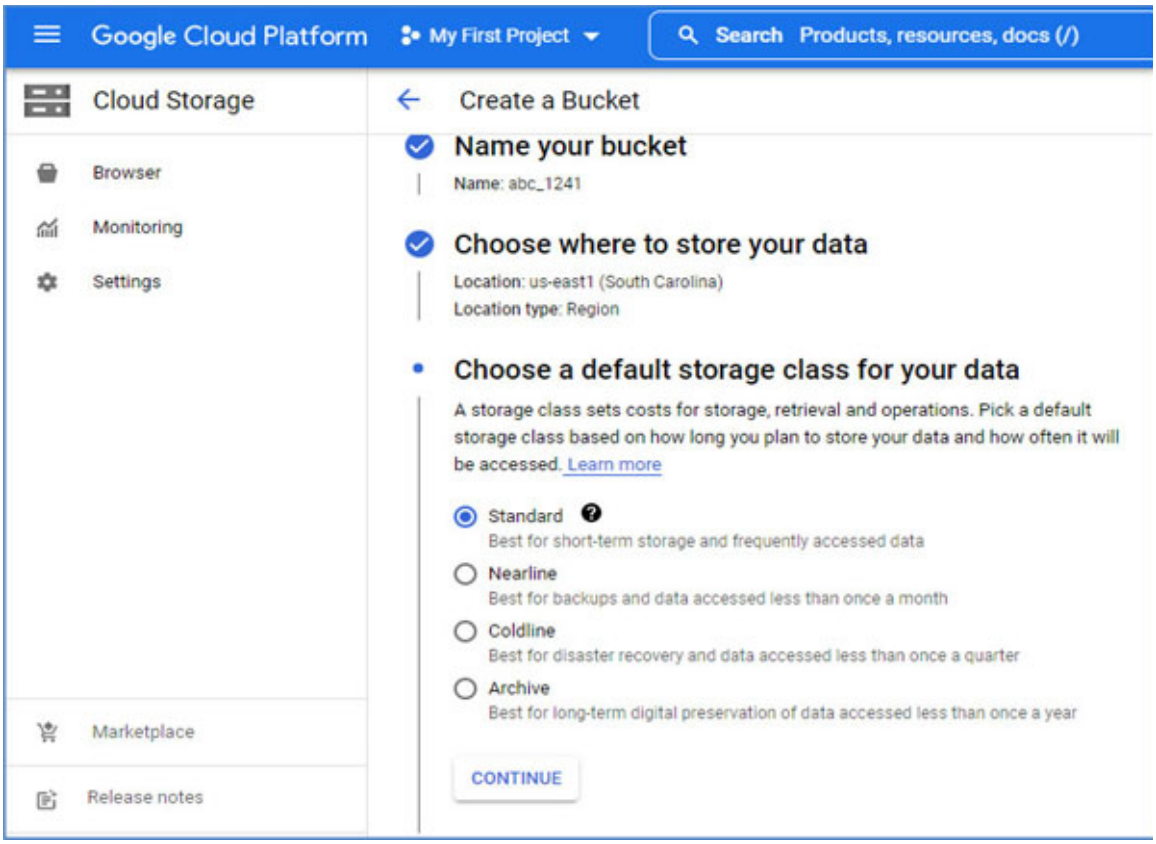

Figure 3.4: *Choosing a default storage class*

5. Click on **`Create`**, as shown in *figure 3.5*:

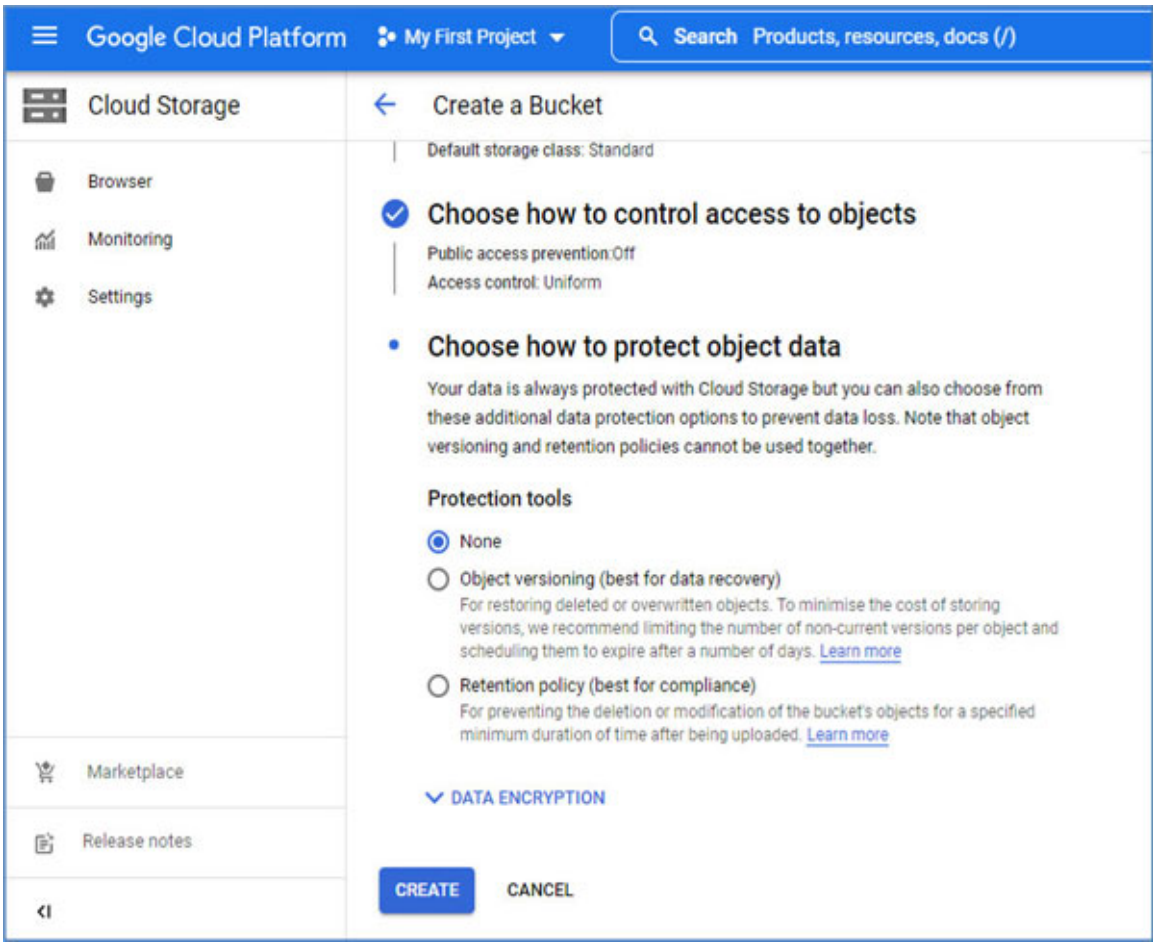

Figure 3.5: *Creating a Google Cloud storage bucket*

Your cloud storage bucket will get created.

6. Now, you will get an option to upload any file to your cloud storage bucket. You can upload some demo files from your local system by clicking on **`UPLOAD FILES`**. Your file should be visible now. Refer to *figure 3.6*:

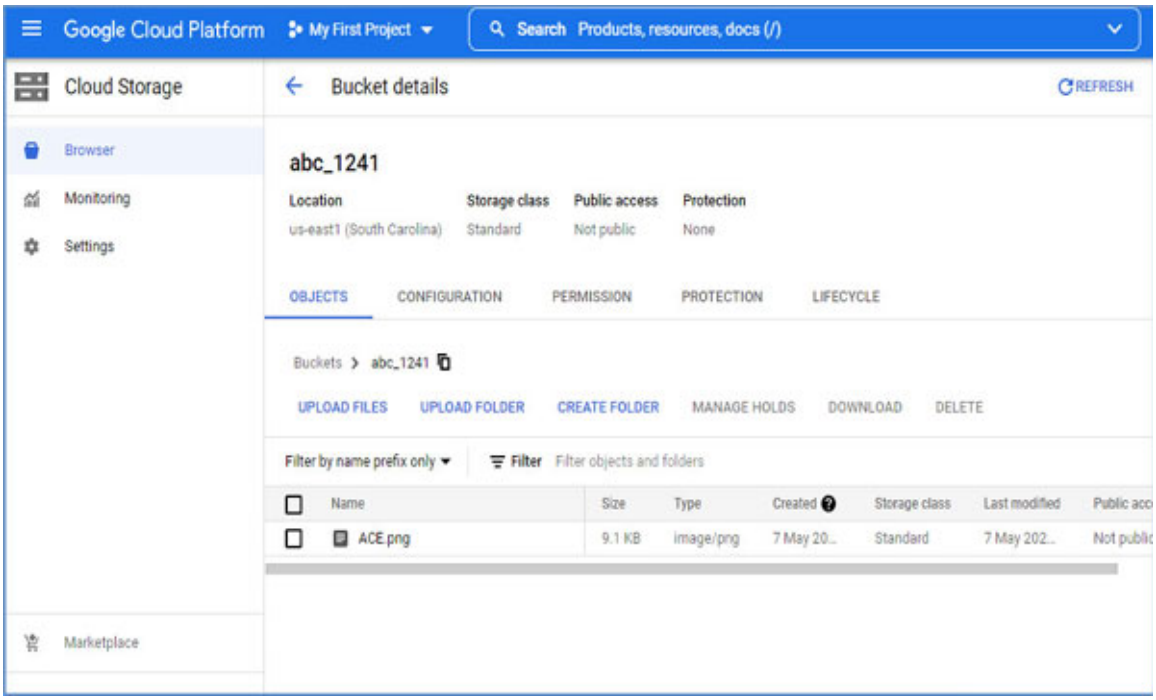

Figure 3.6: *Upload files to the storage bucket*

7. Click on **`Cloud Storage`**. You should be able to view the newly created cloud storage bucket now, as shown in *figure 3.7*:

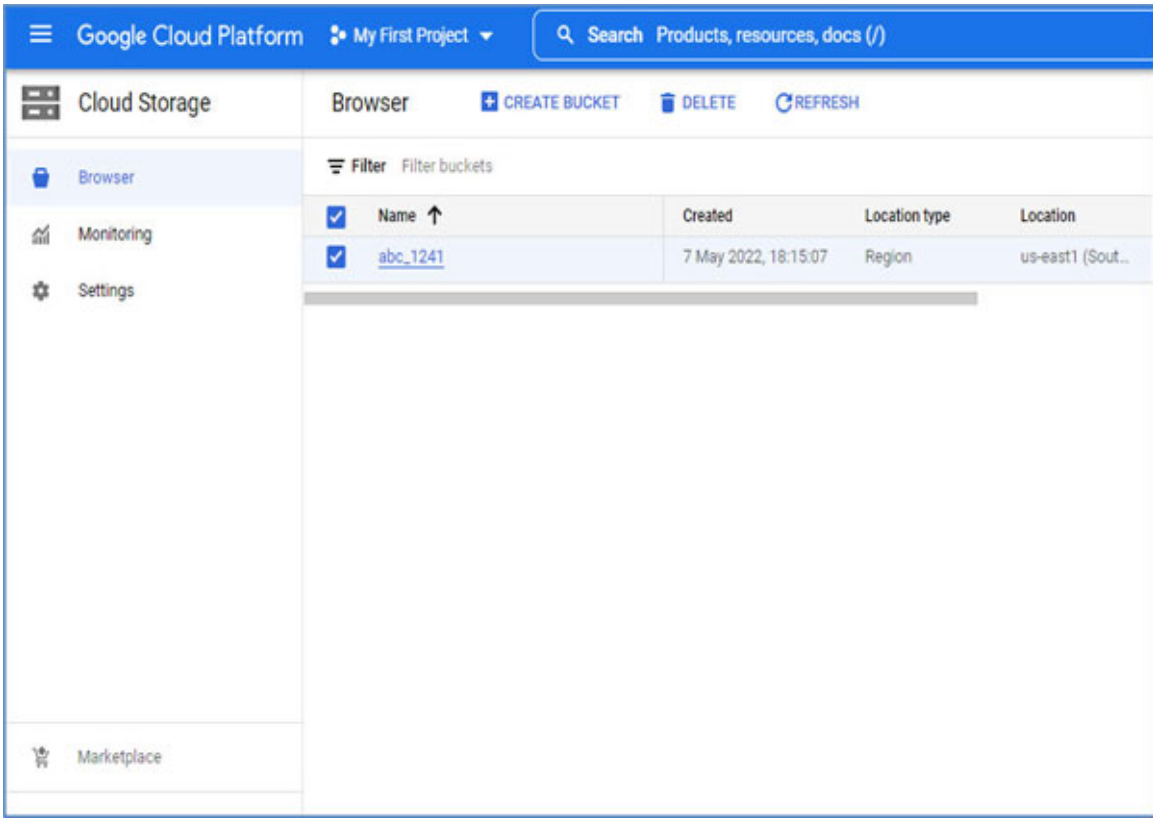

Figure 3.7: *Cloud storage view*

8. Now in order to share this object with other users, click on three dots at the extreme right and then click on **`Copy authenticated URL`**. You can share this URL with other users who want to access this object without the need for any user credentials. Refer to *figure 3.8*:

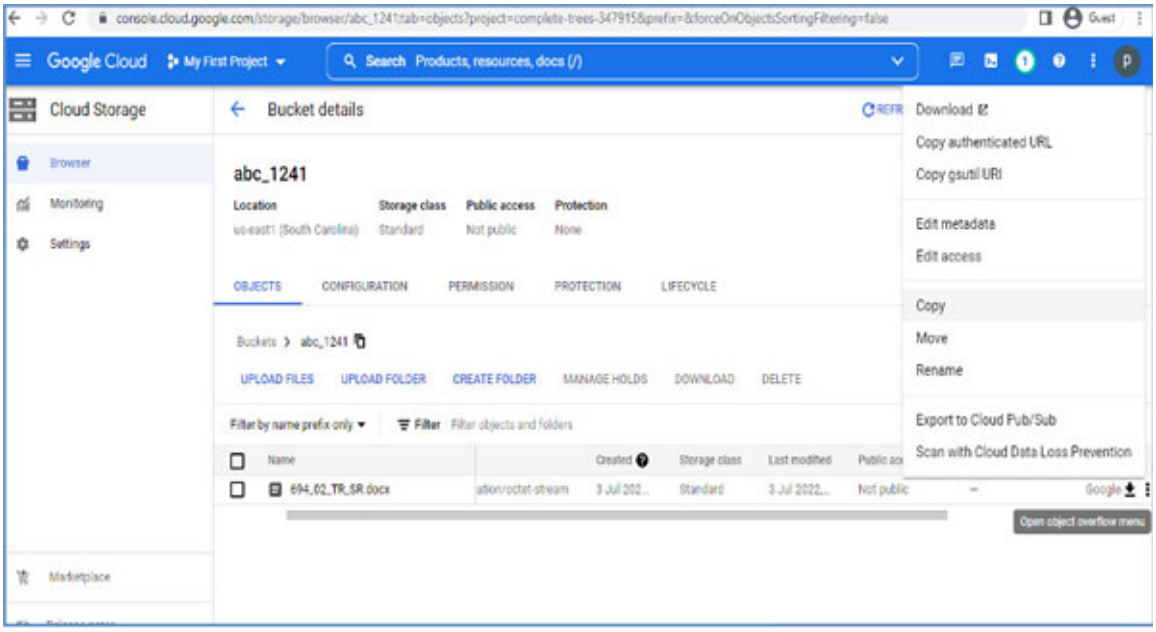

Figure 3.8: *Sharing the object*

Conclusion

In this chapter, we have briefly touched upon all storage options in the Google Cloud Platform. Cloud Storage is being widely used in GCP not only for storing unstructured data like video files or documents but also for backup and archival. Apart from this, Cloud Storage is also used as a data lake for storing raw data that can be processed further and can be used in analytics and AI/ML. While all storage options are equally important from a usage point of view, Cloud Storage has become very popular in recent years due to its wide scope. Readers are advised to go through the GCP site and other online study material to develop further understanding on this. In the upcoming chapter, we will be covering various database services available in Google Cloud. Readers will be able to understand the databases that are supported in the Google Cloud Platform along with the various benefits of using database service offerings available in GCP.

Key terms

Service level agreement (SLA): A contract between service provider and customer that defines the quality, availability, and responsibility of the service provider for the service that is being provided.

- **SAN**: Storage Area Network
- **NAS**: Network Attached Storage
- **NFS**: Network File System
- **CSV:** Comma-separated value

Questions

1. What are the various types of storage?
2. What are various storage services available in GCP?
3. Define various storage classes in Cloud Storage?
4. Give some examples of use cases for various storage classes in Cloud Storage.

Further reading

- **https://cloud.google.com/products/storage**

CHAPTER 4

Database Services in Google Cloud

A cloud database is a database service built and accessed through a cloud platform. It serves many of the same functions as a traditional database with the added flexibility of cloud computing.

-IBM

Introduction

Managing databases and underlying infrastructure has always been a challenging task for IT organizations. Finding the right resource that is capable of creating and maintaining databases has always been critical. Data transformation is a very broad topic, and getting people with the right skill sets has been a tedious task. Considering the current requirement, various cloud service providers have come up with Database-as-a-Service (DBaaS) offerings for SQL and No SQL databases.

Structure

In this chapter, we will cover the following topics:

- Cloud SQL
- Cloud Spanner
- Cloud Bigtable
- Cloud Firestore
- Cloud Memorystore
- BigQuery
- Comparison between various database services
- Lab Exercise

Objectives

After reading this chapter, you will be able to understand the various database service options available in the Google cloud platform. Readers will be able to select the right database service for their needs. At the end of the chapter, there is a lab exercise that will help readers to understand the steps that are required to configure database service for their infrastructure and application needs. Readers are advised to go through other documentation to enhance their knowledge.

Cloud SQL

Cloud SQL is the relational database service provided by the Google Cloud Platform and one of the most widely used services in GCP. Cloud SQL is a highly scalable service that helps the organization with managing the databases. IT teams can focus on developing and updating their applications instead of focusing on managing or maintaining their databases.

Features

As stated previously, Cloud SQL is one of the most widely used services in the Google Cloud Platform. It has many useful features associated with it, some of which are described as follows:

- **Highly availability**: You can create regional, dual region, and multi-region Cloud SQL instances. In a regional Cloud SQL instance, a replica of a SQL instance is managed in other zones to support high availability and scalability.
- **Secure**: Cloud SQL is a secure service. It provides security through multiple layers. Cloud SQL supports encryption for data at rest and in transit. Apart from it, we have firewall and authentication services available to provide additional security. It supports private connectivity with VPC and is compliant with various security standards such as ISO, PCI DSS, and HIPPA.
- **Fully managed service**: Cloud SQL is a fully managed service. It automates backup, replication patch upgrades, and so on. End users need not to be worried about managing their databases.
- **Less cost**: When you compare your cost with the On-Prem system, where you are responsible for creating and managing the underlying infrastructure, patches, and so on, Cloud SQL provides you with cost benefits.

- **Easy to migrate**: Cloud SQL has native tools like Database Migration Service that makes it easy for you to migrate your databases from On-Prem to the cloud or from one cloud to another.
- **Integration**: Cloud SQL can be integrated easily with other GCP services such as Compute Engine, App Engine, Google Kubernetes Engine, and so on.
- As of date, the database size supported by Cloud SQL is 64 TB. It supports 99.95% SLA.

Cloud SQL supports various Database Engines, some of which are as follows:

- **Cloud SQL for MySQL**: Cloud SQL for MySQL service is used when you want to use this open-source database in any of your applications. Application code can be written in Jaya, Python, PHP, GO, Node.js, and Ruby. Even if your applications are On-premise, you can still make use of Cloud SQL services for connecting your front-end applications to the Cloud SQL instance in the GCP Cloud. As ofdate, Cloud SQL for MySQL supports 624 GB of RAM and 96 vCPUs. You can connect to Cloud SQL for MySQL using various services such as App Engine, Cloud Function, Cloud Run, Google Kubernetes Engine, and so on. You can export and import MySQL database using the `mysqldump` command. You can make use of automated backup and on-demand backup of Cloud SQL service.
- **Cloud SQL for SQL Server**: Cloud SQL for SQL server is another database management engine offered by GCP. Microsoft SQL server is a database server created by Microsoft and is widely popular among database administrators and application developers. This is a fully managed service where the Google cloud platform takes care of the management of the SQL server. It allows users to focus more on database creation and management. Like Cloud SQL for MySQL, this service also gives you the option to create automatic and on-demand backups. You can connect with Cloud SQL for SQL servers using App Engine, Cloud Function, Cloud Run, Google Kubernetes Engine, and so on.
- **Cloud SQL for PostgreSQL**: PostgreSQL is a widely popular, reliable, open-source relational database that supports relational (SQL) and non-relational (JSON) queries. It is considered as an alternate to the Oracle database due to its rich features and Open-Source approach. Cloud SQL for PostgreSQL is another managed service that helps you to manage and

maintain the PostgreSQL database in the Google cloud Platform. It also supports 624 GB of RAM and 96 vCPUs custom machine type. You import and export databases using SQL dump and take automated and on-demand backups of the database like other Cloud SQL offerings. We can also connect to Cloud SQL for PostgreSQL using various GCP services such as App Engine, Cloud Functions, Cloud Run, Google Kubernetes Engine, and so on.

Cloud Spanner

Cloud Spanner is a horizontally scalable, highly consistent, and relational database management service offered by the Google Cloud Platform. It supports global online transaction that supports more then 65 TB of database size. Unlike Cloud SQL, you need not to create replicas in Cloud Spanner as it automatically manages replicas. Cloud handles automatic sharding that helps it in becoming horizontally scalable. It also supports multiple programming languages such as C#, C++, Go, Java, Node.js, PHP, Python, Ruby, and so on. and provides 99.999% SLA. It also provides synchronous replication across regional and multiregional locations. Cloud Spanner is expensive, and hence, it is used for specific use cases where you require high availability for critical workloads. Some of the use cases of Cloud Spanner are Finance companies, the Gaming Industry, or the Retail Industry. A dedicated Google network helps Cloud Spanner in enriching with its features and functionalities.

Cloud Bigtable

Bigtable is a fully managed, No-SQL database service offered by the Google Cloud Platform. This is a highly scalable, low latency, distributed, and feature-rich service, which is used for storing non-relational data. You can add additional nodes dynamically to scale the performance of Cloud Bigtable. It provides a regional and multiregional presence with an automatic replication feature that adds up its availability to 99.999%. This service can easily integrate with other data services such as dataproc and dataflow. Cloud Bigtable is a columnar database that stores data in the key-value format and can widely be used for analytical workload requirements. It supports a petabyte of data. It can scale dynamically by adding or removing nodes to the cluster. One can also use multiple clusters. Bigtable separates compute and storage, and that helps it in scaling so dynamically. Google services like Google maps, Search, Maps and Gmail, and so on make use of this service.

Cloud Firestore

Firestore is a feature-rich, managed, and scalable serverless document database service available in the Google Cloud Platform. Cloud Firestore does not store data in rows and columns; instead, it stores it in documents. Documents contain key-value pairs. A collection is a combination of multiple documents. Firestore is schemaless, which means you can add additional fields to it. For example, a "class" can be considered as a collection that contains various documents such as "SectionA", "SectionB", "SectionC", and so on. Now, these documents can have various fields inside them, such as "Student Name", "Student Age", and so on. These fields are considered as keys and can have values associated with them. Firestore syncs automatically as per the user demands. It supports automatic multiregional replication. It supports 99.99%–99.999% availability SLA. One of the major benefits of using cloud Firestore is its cost. Here you are charged only for the data access, network, and storage. There are no sizing constraints mentioned by Google in its documentation, and hence, it can grow up to any size. Firestore has an export and import service. You get an option to export all documents or individual collections.

Cloud Memorystore

Google Cloud Memorystore is another database service provided by the Google Cloud Platform. This is an in-memory database service that uses a memory cache to store and process transactions. Because of this, cloud, Memorystore is used where we have very low latency requirements for the transactional workload. It provides sub-millisecond data access capability. It is a managed service for Redis and Memcached open-source databases and provides high availability and failover capabilities. It provides 99.9% availability SLA. Memorystore is primarily used for in-memory caching and used in conjunction with other database services for a faster read. Whenever a read request is made, if it is a cache hit, data is read from the Memorystore, and if it is a cache miss, then the data is read from the actual database. Multiple read replicas are created for fulfilling faster processing and high availability requirements.

BigQuery

Google Cloud BigQuery is a serverless data warehouse service that is used primarily for analytics. BigQuery is one of the most popular services in the

Google cloud Platform. Like other data warehouse services, BigQuery uses a columnar database for storing the data, making it suitable for the write intensive workload. BigQuery is suitable for **Online analytical processing** (**OLAP**) and is not preferred for **Online transactional processing** (**OLTP**) requirements. It is considered as highly scalable and cost-effective service provided by the Google Cloud Platform.

In the present age of digital transformation, when most of the companies want to make use of analytics in order to sustain and grow, data warehouse services like BigQuery plays a very important role. Data coming from various sources can be transformed and ingested into BigQuery. Once the data has been stored, we can run interactive queries on it. Either you can create a dataset or import datasets and perform SQL-like queries in order to get more insight into this data. Later on, you can make use of visualization tools like data studio or Looker by integrating them with BigQuery. BigQuery output can also be used to create machine learning models that can be used for performing predictive analysis.

Features

The following are some of the features of BigQuery:

- **Easy to integrate**: BigQuery is easy to integrate with many other services such as Dataflow, Dataproc, Data Fusion, Cloud Storage, vertex AI, Data Studio, Looker, and so on. It can also get integrated with external services like the Hadoop cluster.
- **Serverless**: BigQuery is a serverless service; therefore, all the provisioning is handled by Google, and users can focus on data instead of infrastructure provisioning.
- **BigQueryML**: BigQuery ML is a machine learning service offered by BigQuery that can be used to create and manage machine learning models using SQL queries. In case if you are using BigQuery ML, you do not need to move the data to some other service in order to build ML models.
- **Real-time analytics**: BigQuery allows you to perform real-time analysis by integrating itself into the ETL pipeline. You can build ETL pipelines for streaming data coming from IoT sensors or other sources that can be consumed by BigQuery for analytics.

- **Extremely fast**: BigQuery is extremely fast. It uses Google's colossus file system to store the data, which uses fast and distributed storage. Compute is Dremel which converts SQLqueries into execution trees. Apart from it, BigQuery makes use of Google's Jupiter network to move data between compute and storage.
- **Easy to use**: BigQuery is easy to use as it uses standard SQL queries on the datasets. Anyone with a fair understanding of running SQL queries can make use of this service.

Comparison between various database services

Table 4.1 discusses the differences between various database services:

Cloud SQL	Cloud Spanner	Cloud Firestore	BigQuery	Cloud Bigtable
Relational databases services used for Web framework	Horizontally scalable, high-performing relational database	Document database used for mobile/web applications	Data Warehouse	Heavy reads/writes requirements, NoSQL database
Suitable for ERP, eCommerce applications	AdTech, Fintech, and so on, Global supply chain	eCommerce, Game state, IOT	Analytics requirements	AdTech, Fintech and so on
Structured, Supports SQL	Structured, Supports SQL	Structured, No SQL, Key-value structure	Structured, Supports Sql like queries	Structured, NoSQL
Terabytes	Petabytes	No constraints	Petabytes	Petabytes
99.95% availability	99.999% availability	99.999% avaiability	99.99% availability	99.999%b availability

Table 4.1: *Comparison between various DB services in GCP*

Lab exercise

Let us now work on a few lab exercises.

Creating Cloud SQL MySQL instance and connecting it with a virtual machine

To create a Cloud SQL MySQL instance and connect it with a virtual machine, follow the following steps:

1. Login to your GCP console and click on the **Navigation menu**. Under **Compute Engine**, click on **create VM instance**. Give the compute engine a unique name and select the preferred region and zone. Keep Series and machine type as it is. Refer to *figure 4.1*:

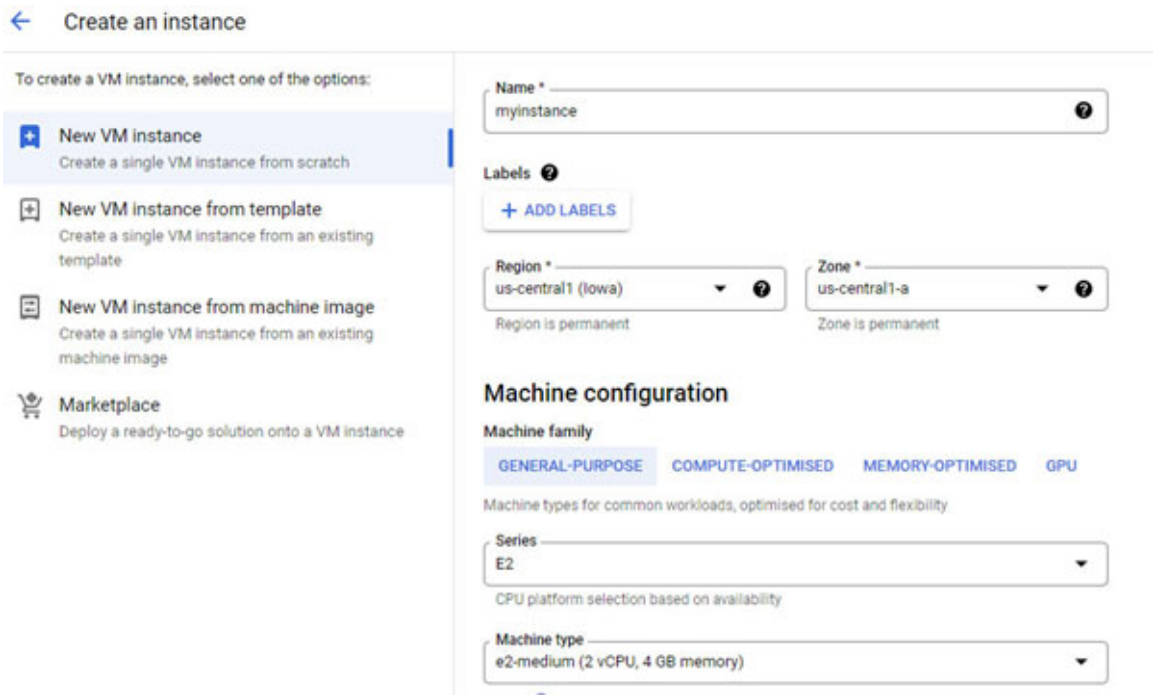

Figure 4.1: *Creating a VM instance*

2. Let the boot disk remain as Debian GNU/Linux 11 and click on **`Networking, Disks, Security, Management, Sole tenancy`**, as shown in *figure 4.2*:

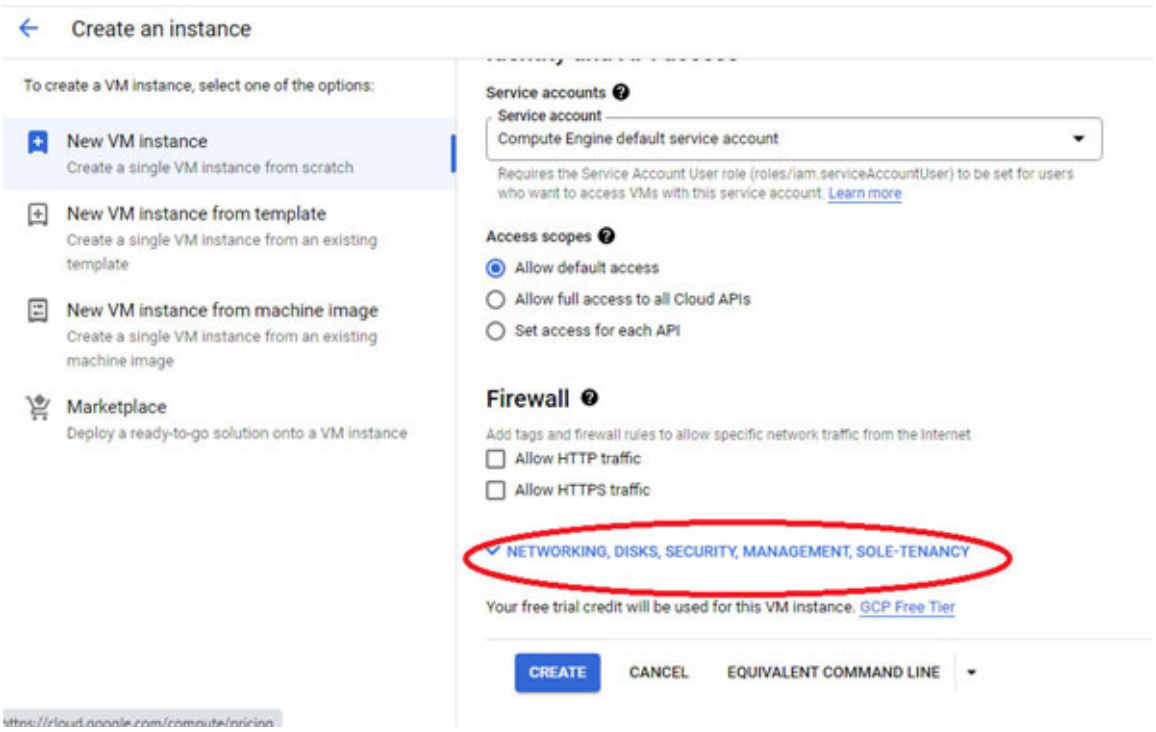

***Figure 4.2**: Option networking, disks, security, management and sole tenancy under New VM instance*

3. Now, we will assign a static external IP address here. Click on the down arrow next to **Network Interface**, as shown in *figure 4.3*:

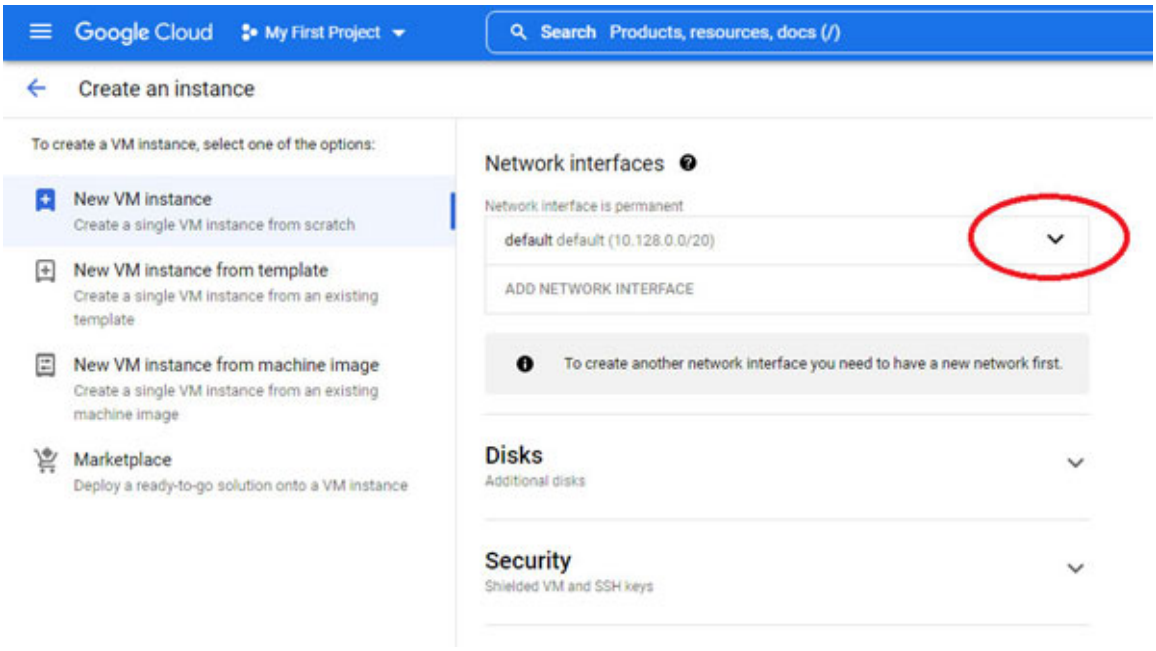

***Figure 4.3**: Assigning a static external IP address*

4. Click on External IP and then click on **`Create IP Address`**. Refer to *figure 4.4*:

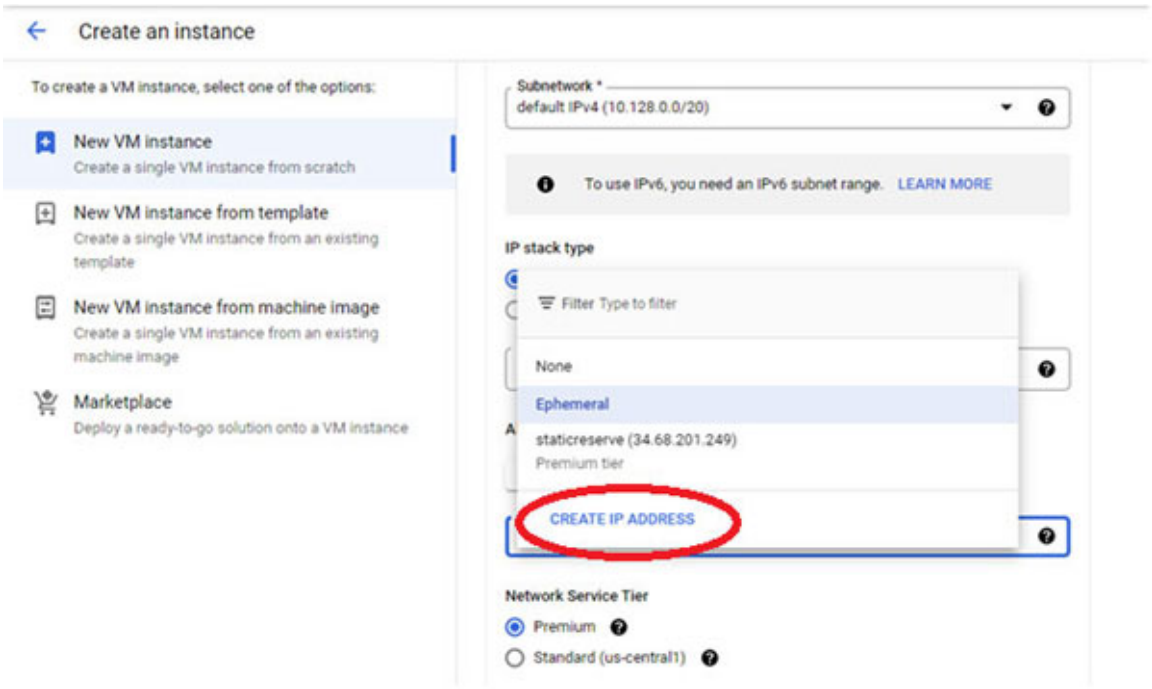

Figure 4.4: *Creating an IP address*

5. Give a name and **reserve** a static IP address, as shown in *figure 4.5*:

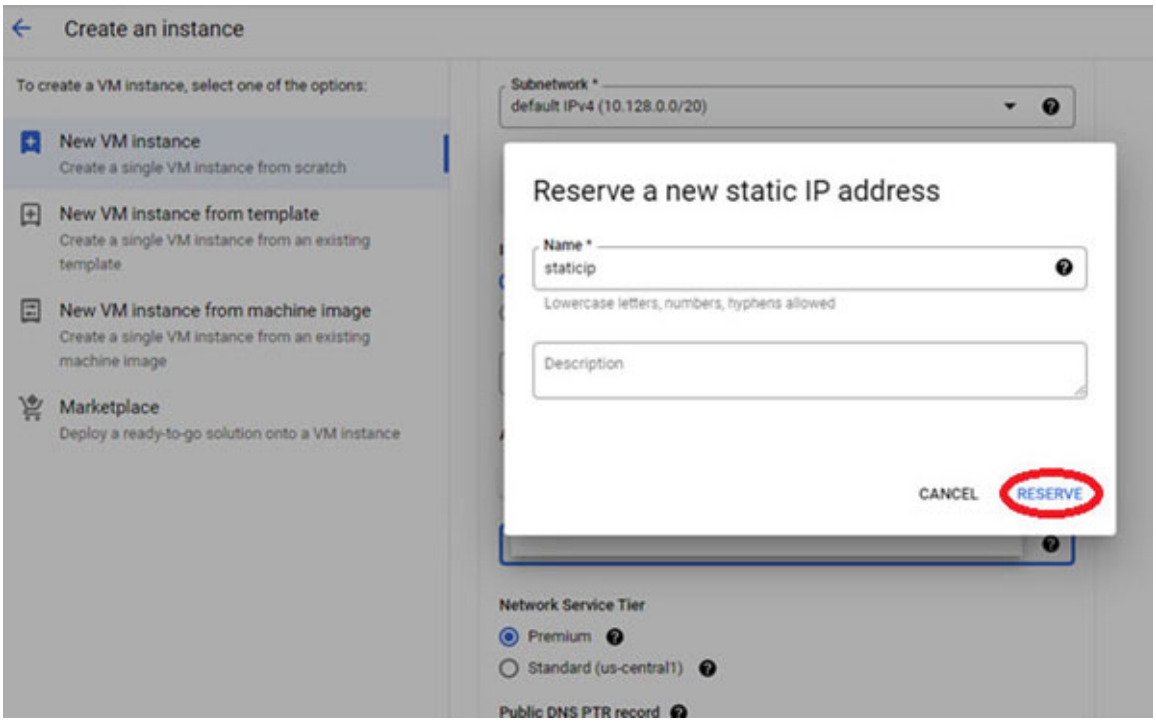

Figure 4.5: *Giving a name and reserving a static IP address*

6. Click on **`Done`** and then **Create**. Wait for a few minutes, and your new VM instance will get created, as shown in *figure 4.6*:

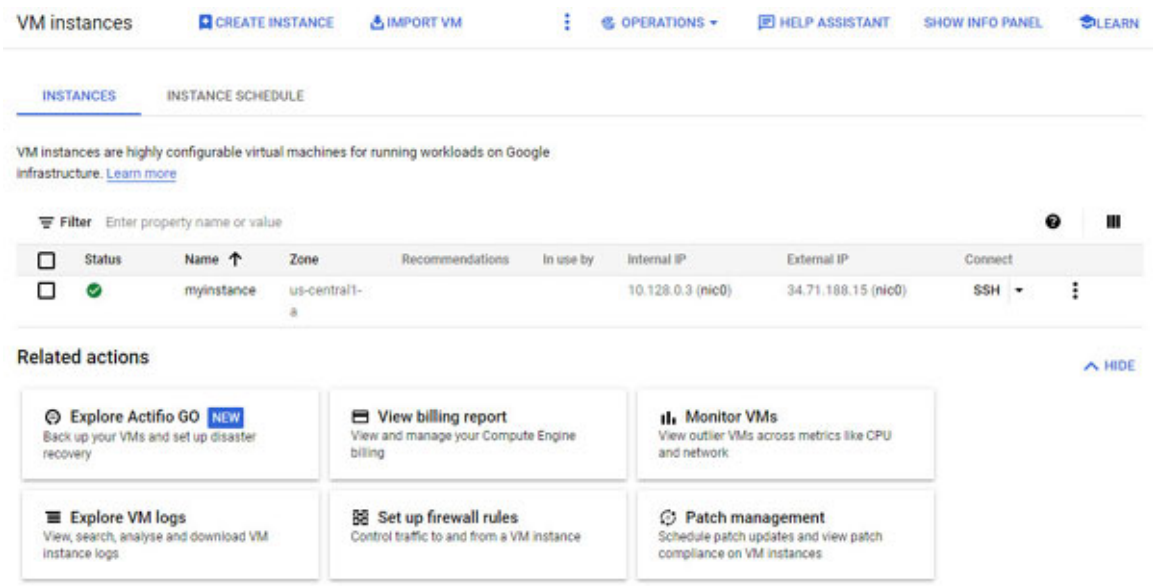

Figure 4.6: *Successfully creating a new VM instance*

7. Now, we will go ahead and create a Cloud SQL for the My SQL DB instance. Go to the Navigation menu and select SQL from there. Click on **`Create Instance`**, as shown in *figure 4.7*:

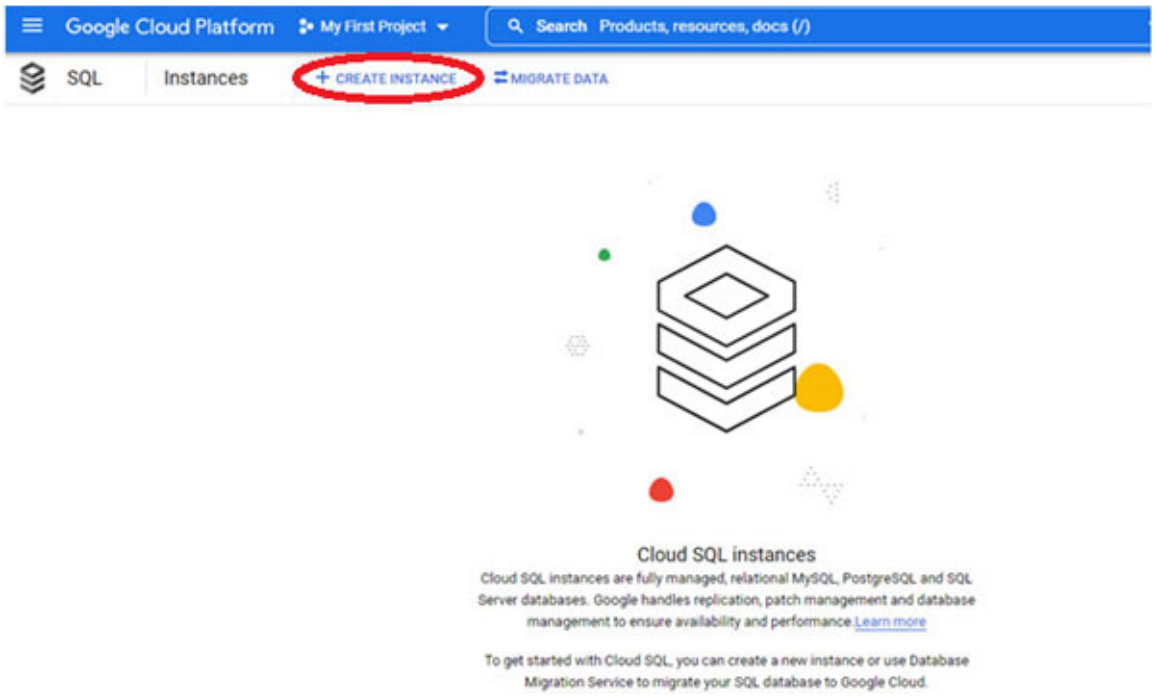

Figure 4.7: *Creating a new Cloud SQL*

8. Now, under MySQL, please select **`Choose MySQL`**, as shown in *figure 4.8*:

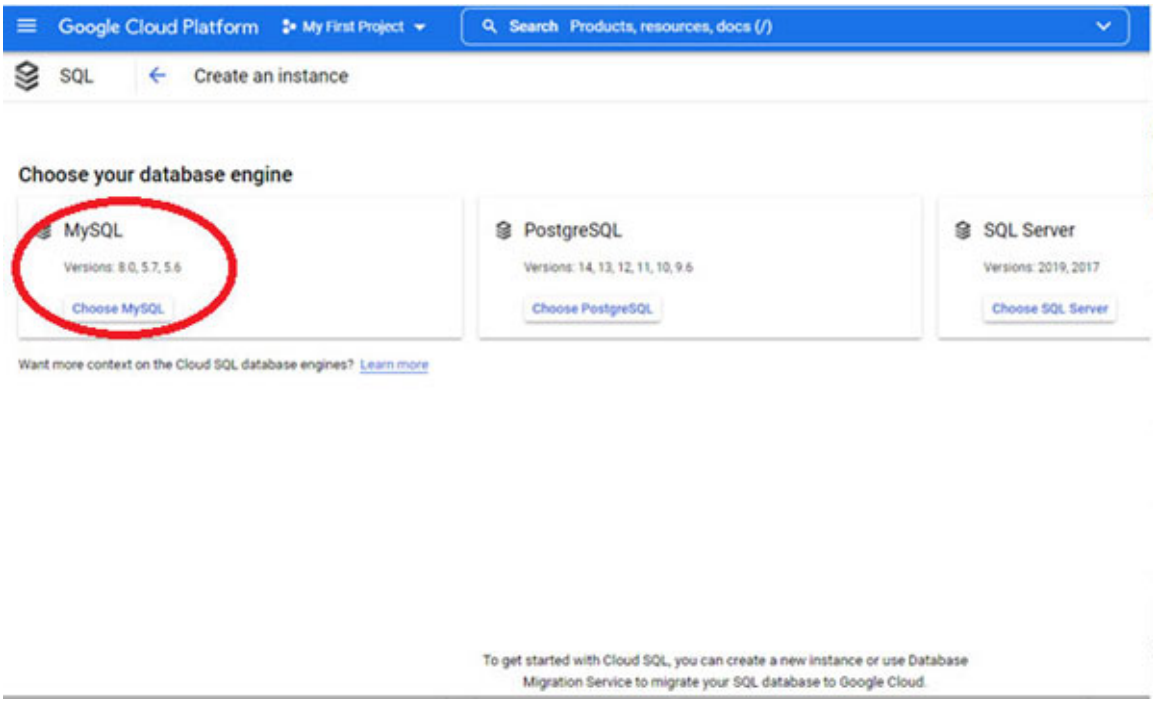

Figure 4.8: *Choose the MySQL option*

9. Please select the instance ID and the password for the database instance that we are trying to create. Select the database version and the

environment. Refer to *figure 4.9*:

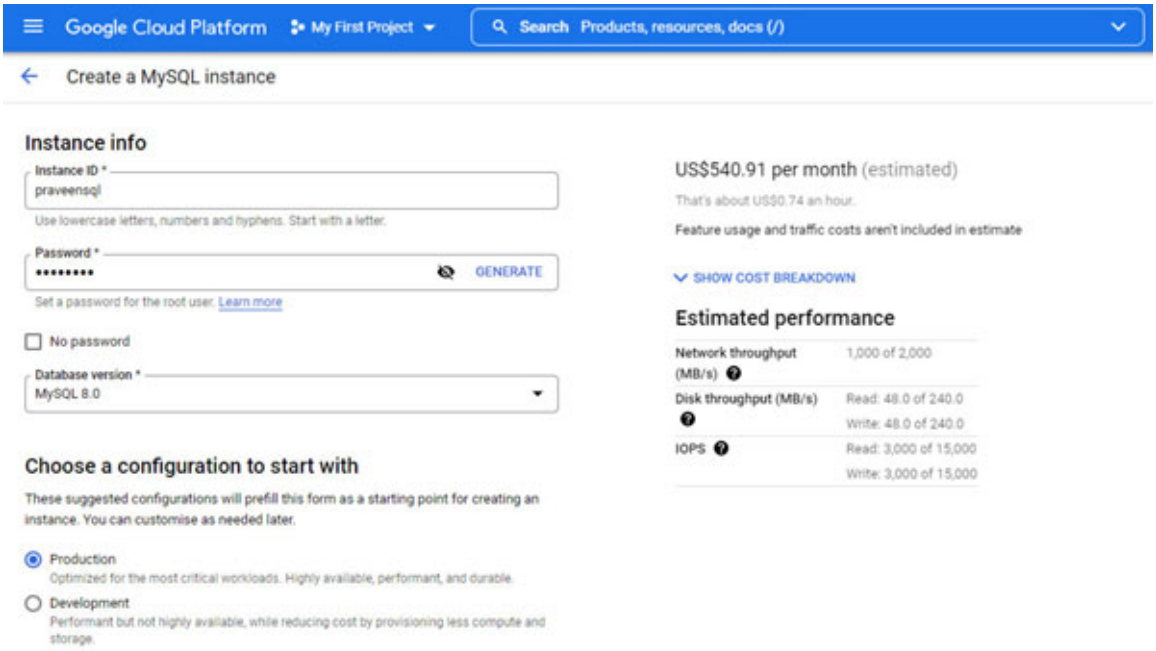

Figure 4.9: *Fill in the details*

10. Select Region and Multiple zones. If you do not want to create a multiple zones instance, you can always select a specific zone. Also, please ensure that the selected region is the same as the region selected for the VM instance that has already been created. Click on **`Create Instance`**. Refer to *figure 4.10*:

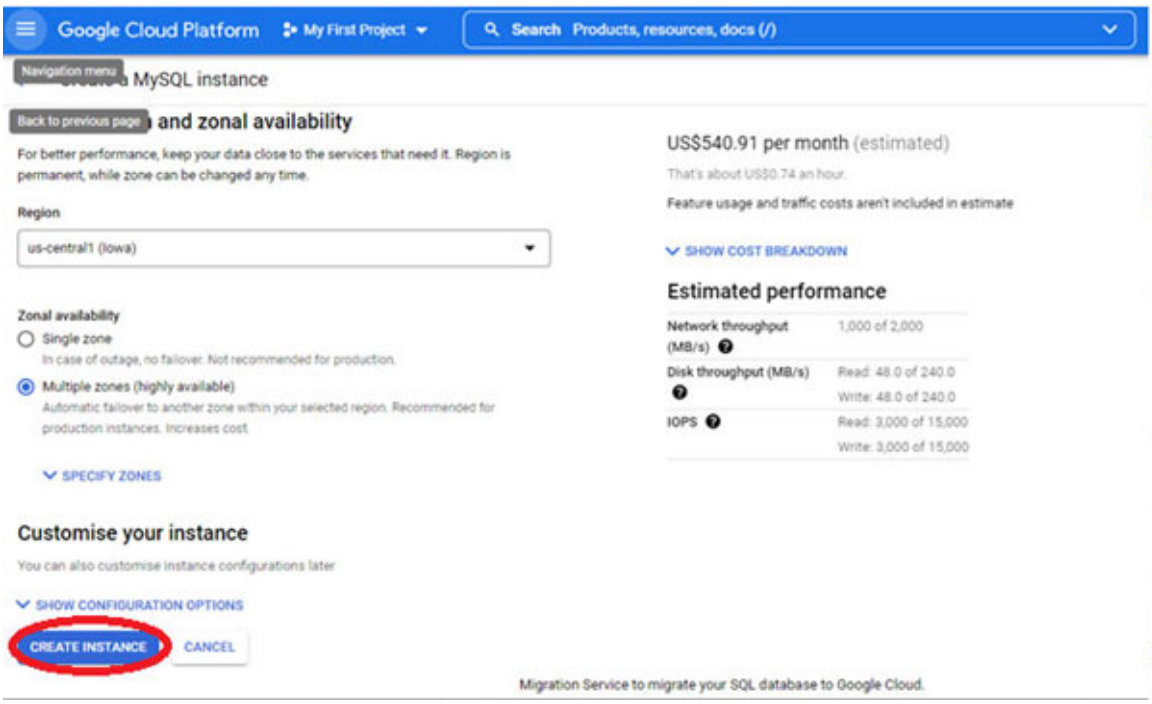

Figure 4.10: *Selecting region and multiple zones and creating the instance*

11. A new Cloud SQL instance will get created. Please make a note of the public IP, as we will require it in the upcoming steps. Refer to *figure 4.11*:

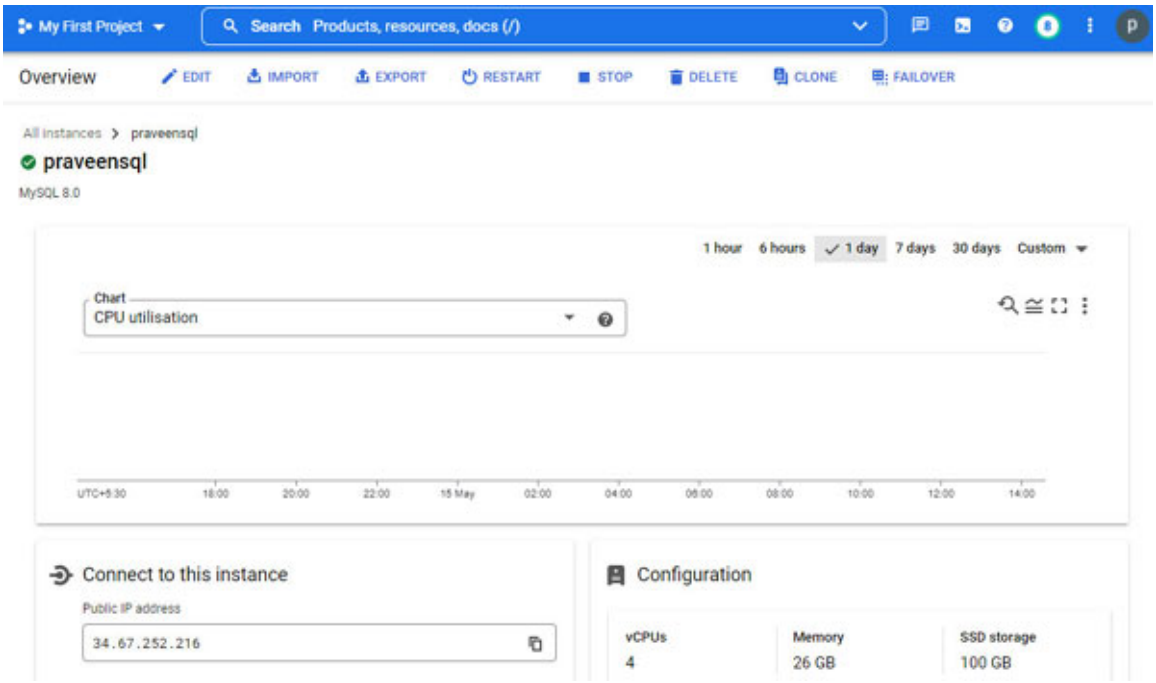

Figure 4.11: *New Cloud SQL instance is created*

12. Now, click on the **`Connections`** tab in the Navigation menu. Now, click on **`Add Network`**, as shown in *figure 4.12*:

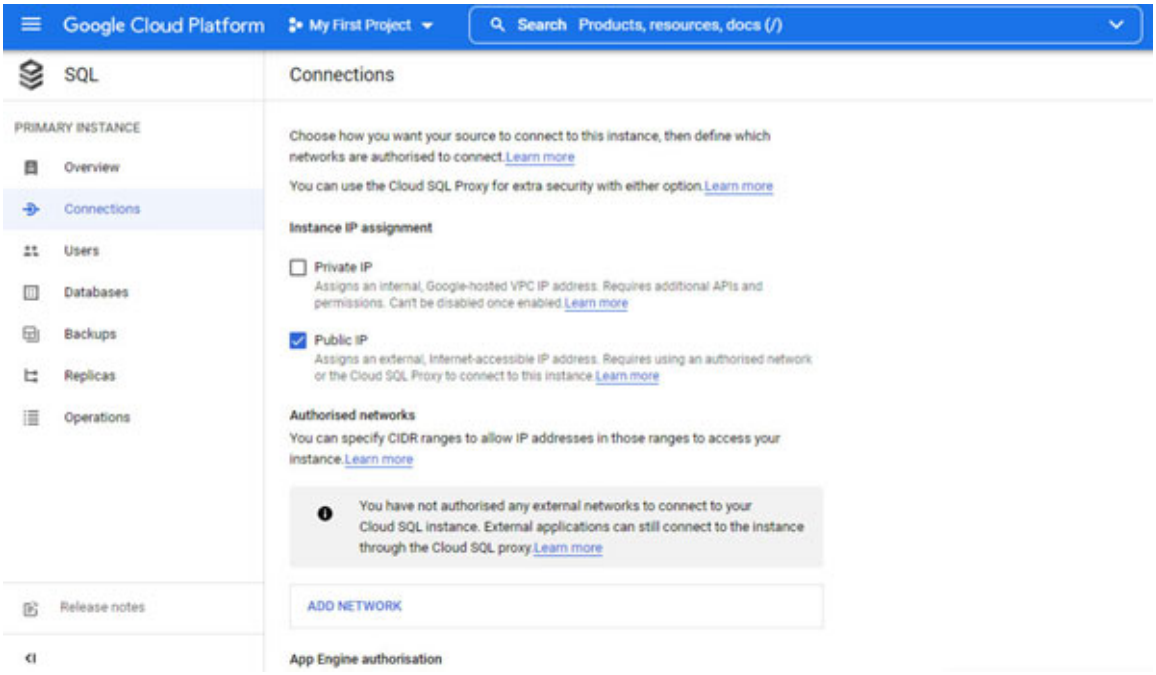

Figure 4.12: *Add the network*

13. Copy and paste the static IP address of the VM instance that we have created already. Click on **`Done`** and then click on **`Save`**. Refer to *figure 4.13*:

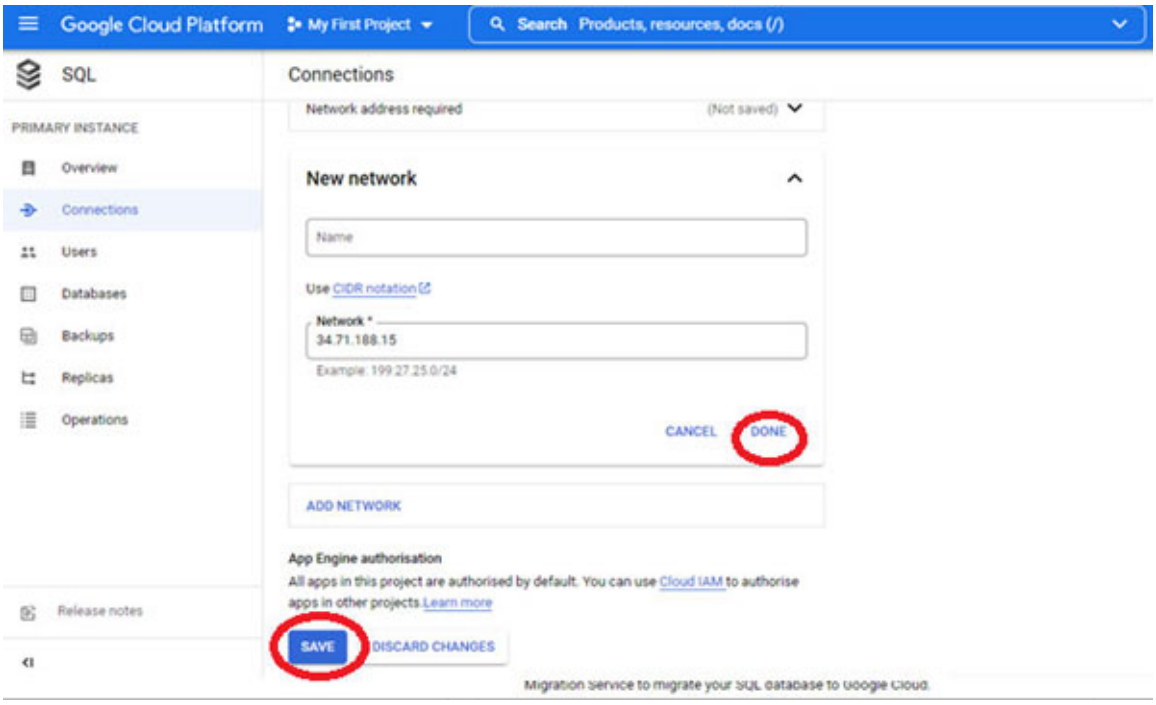

Figure 4.13: *Save the project*

Testing the connectivity between the VM instance and Cloud SQL instance

Now, we will test connectivity between the VM instance and the Cloud SQL instance.

1. Go back to the VM instance that has already been created and click on SSH. Refer to *figure 4.14*:

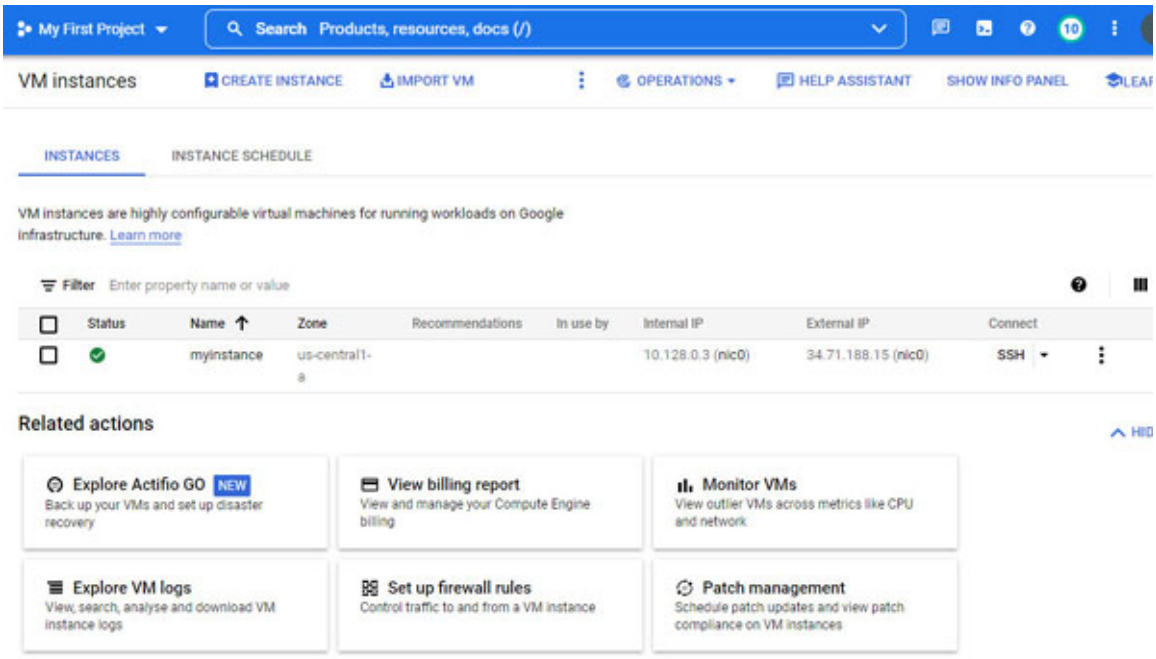

Figure 4.14: *Select SSH under connect option*

2. We will install the MySQL client in the system by running the following commands:

```
sudo apt-get update
sudo apt-get install mysql-client -y
```

   Refer to *figure 4.15*:

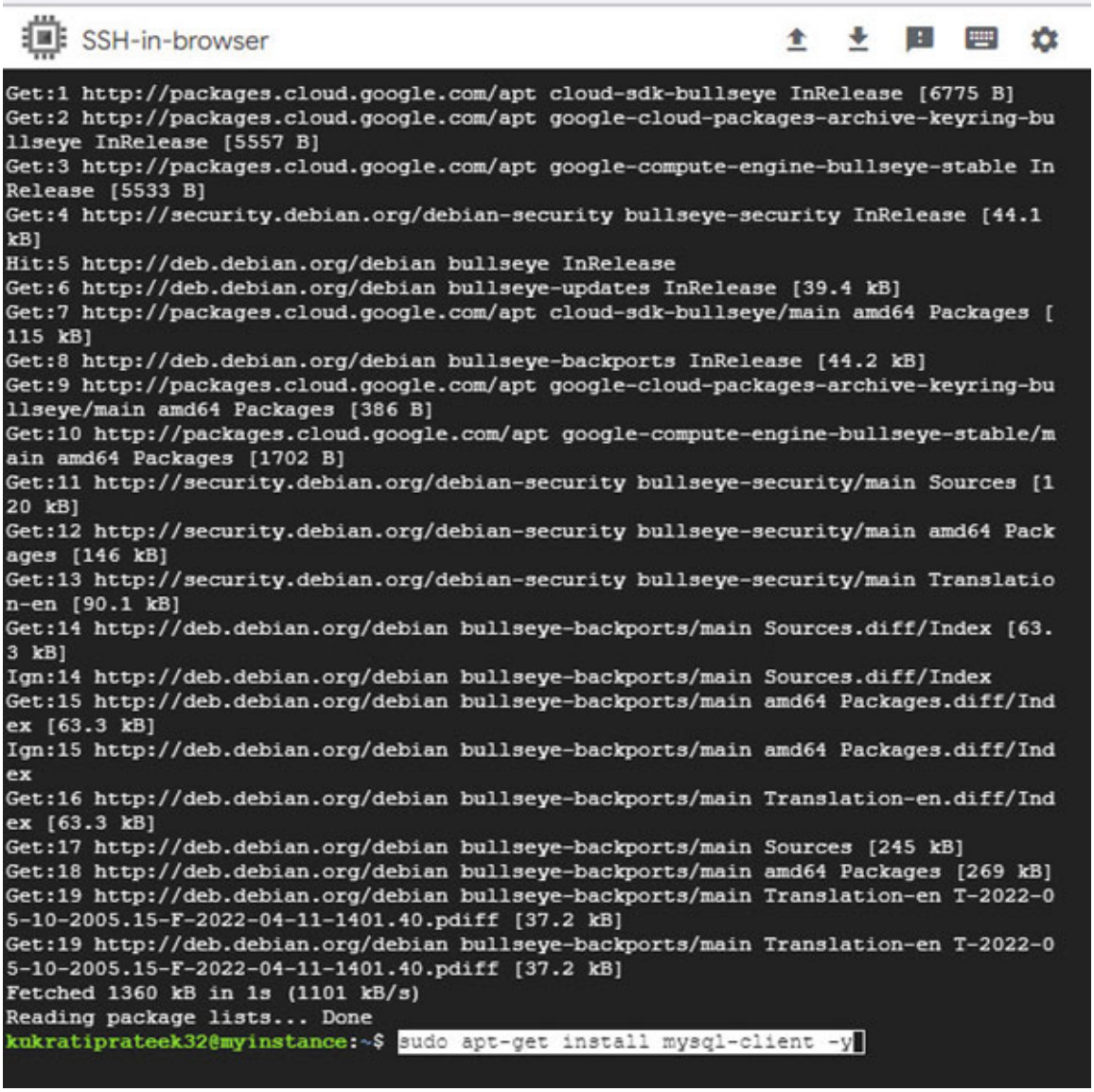

Figure 4.15: *Command to install MySql client in the system*

3. Now let us connect to the Cloud SQL instance by using the following command. Replace this IP with the MySql instance IP that has been assigned to your instance and enter the password once prompted.

   `mysql – host=35.202.255.15 – user=root – password`

4. Post this, and you should be able to connect to the Cloud SQL instance, as shown in *figure 4.16*:

Figure 4.16: *Connecting to Cloud SQL instance*

Conclusion

In this chapter, we have discussed about various database services provided by the Google Cloud Platform. There is no need to worry about managing the underlying infrastructure, securities, patches, and so on. Databases are scalable and can withstand high workloads by implementing the right strategies. You also have the option to install databases on virtual machines using persistent disks. In this chapter, we have touched upon various offerings at a very high level. Readers are advised to go through GCP documentation links and other videos to develop further understanding.

Key terms

- **OLTP**: Online transactional processing
- **OLAP**: Online analytical processing
- **Scale up**: Increasing compute and processing of an instance.
- **Scale out**: Adding more nodes to a cluster.
- **Data warehouse**: Data warehouse is a system in which you take your transactional data and store it for analysis and Business intelligence.

Questions

1. What is a database, and why it is used?
2. What are various database engines available in GCP?
3. What is the difference between Cloud SQL and Cloud Spanner?
4. What is a document database, and why it is used?

5. What is the difference between relational and non-relational database?
6. What is a data warehouse service provided by GCP, and why it is used?

Further reading

- **https://cloud.google.com/products/databases**

CHAPTER 5

Networking in Google Cloud

Cloud networking offers connectivity to and between applications and workloads across clouds, cloud services, on-premises data centers, and edge networks. It's vital to performance, security, and efficient management of hybrid cloud and multicloud environments.

-CISCO

Introduction

It will not be an exaggeration to state that the network is like the veins of any IT infrastructure through which all components talk to each other. Like other services, it is not an easy task to build and manage any network infrastructure. Therefore, cloud service providers have come up with many services around the network that are easy to manage and very reliable. Most of the GCP network services can be identified by using their names, as they are common for any on-prem infrastructure as well.

Structure

In this chapter, we will cover the following topics:

- Google's global network
- Organization, projects, folders, and resources
- Virtual Private Cloud
- Connectivity options in VPC
- VPC creation best practices
- Cloud NAT
- Cloud Load Balancing
- Instance groups

- GKE Ingress controller
- Hybrid cloud concepts
- Cloud CDN and edge locations

Objectives

After reading this chapter, you will be able to understand the various network service options available in the Google Cloud Platform. Readers will be able to decide which network service should be used for any specific requirement. It will help them in developing a solution for various use cases. Readers are advised to go through other documentation to develop in depth understanding of any specific network service in GCP.

Google's global network

Google Cloud Platform is a global network where all regions and zones are connected with each other by undersea cables. Google cloud network is very efficient as all other Google services, such as YouTube, Gmail, and so on, have been using the same network. Google Cloud provides a very fast, efficient, and reliable network to its users. As of date, there are 35 regions and 106 zones in GCP that are using this network to enable communication between various services.

Figure 5.1: *Google's global network.*
Source: *https://cloud.google.com/about/locations#regions*

Organization, projects, folders, and resources

In order to understand how a virtual private cloud is designed in the Google Cloud Platform, we need to understand the concept of the resource hierarchy. As the name suggests, all objects in the resource hierarchy are connected through parent–child relationship. Top-level parent in this hierarchy is called an organization, and it is usually represented by the company name for which Google Cloud services are being deployed. Under organization, you can have different folders and subfolders, which can be represented by different departments within an organization. For example, an organization can have multiple departments, such as HR, Finance, Marketing, and so on, which can be further divided into multiple projects. Projects can either be different sub-departments or environments such as prod, test, dev, and so on. Projects can be used to separate the resource billing for the resources that are being consumed and for setting up the access permissions for the resources underneath it. Under projects, you have resources that have been provisioned. We will try to understand Identity and Access Management and its implementation strategy around the resource hierarchy in the upcoming *Chapter 6, Security and Monitoring*

Services in Google Cloud Platform. Please note that while organization and folder are not mandatory to provision resources however it is mandatory to the create the project for provisioning them.

Figure 5.2 features the Organization Structure in Google Cloud Platform:

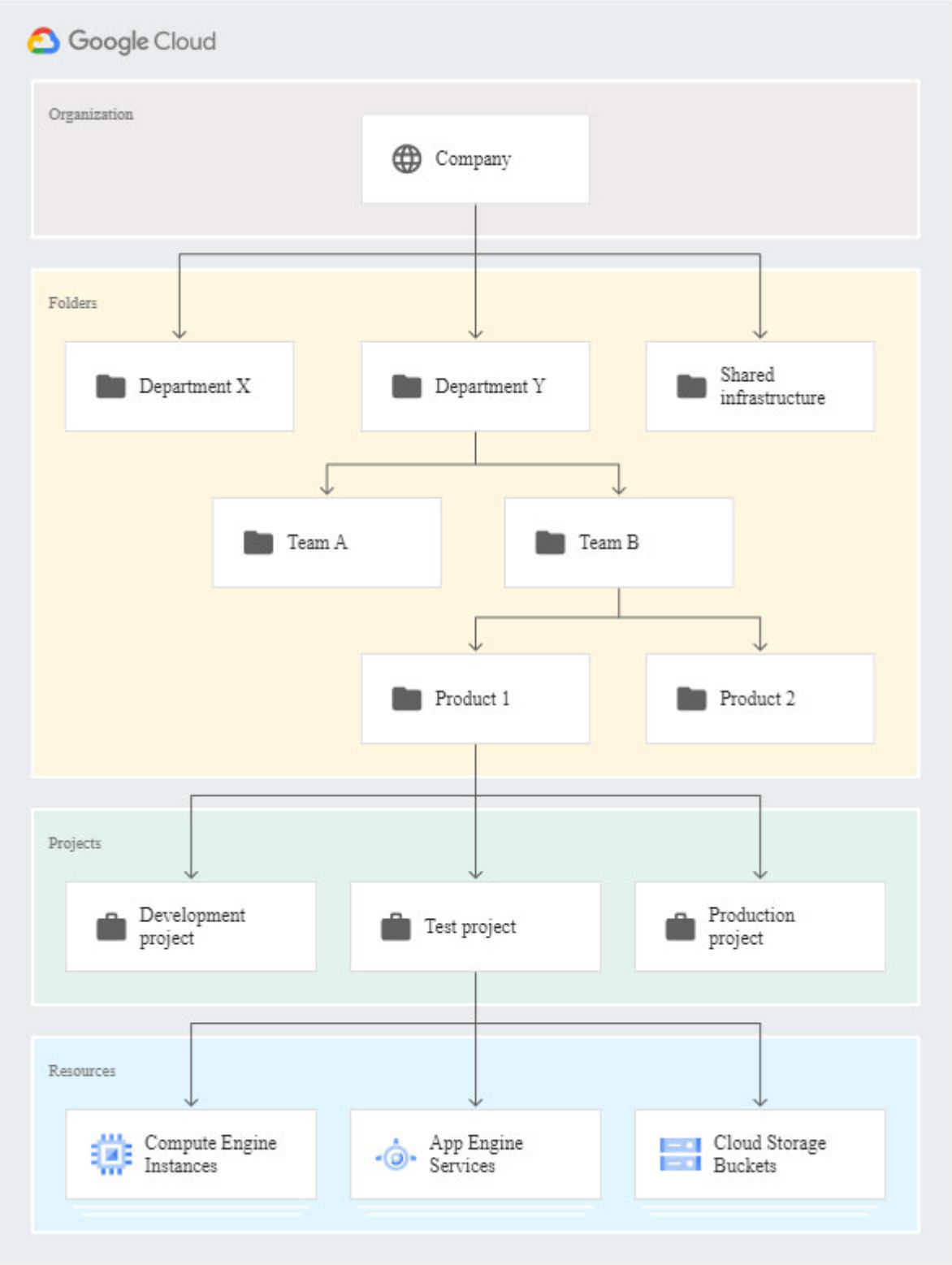

Figure 5.2: *Organization structure in Google Cloud Platform.*
Source: *https://cloud.google.com/resource-manager/docs/cloud-platform-resource-hierarchy*

Virtual Private Cloud

As per Google documentation, "*A Virtual Private Cloud (VPC) is a global private, isolated virtual network partition that provides managed networking functionality for your Google Cloud Platform (GCP) resources*."

VPC is analogous to your physical network, where all resources created within a Virtual Private Cloud can communicate with each other using private IP addresses. **A VPC is a global resource**. **Subnets are a regional resource, and a VPC can have subnets in one or more regions. A VPC can also have multiple subnets in the same region.** Architecture of a Virtual Private Cloud and its underlying components can be understood using *figure 5.3*:

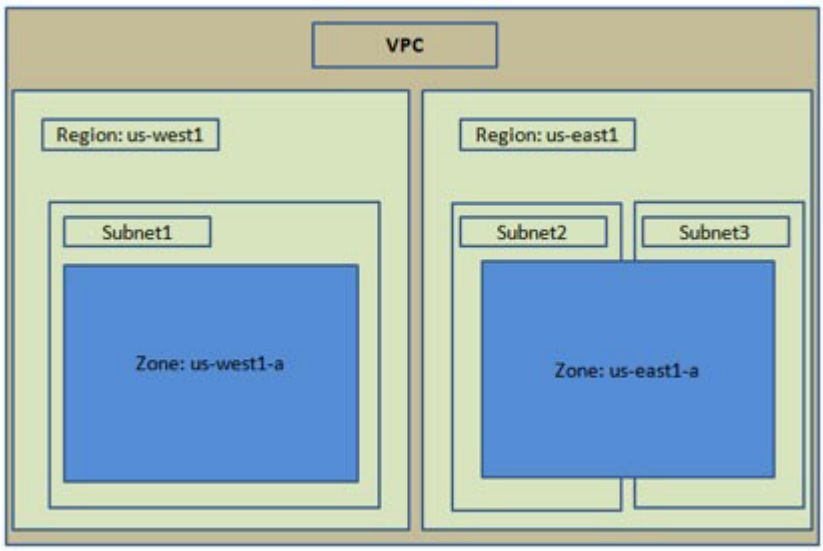

Figure 5.3: *High-level overview-VPC*

It is important to understand that each project has a default VPC, but it can have more than one VPC. Each VPC inside the project is also completely isolated from the others. When we create a project, we do not get any option to create VPC. An auto-mode VPC is already present (pre-created). A default VPC has one subnet for each region. You can create your resources inside these default subnetworks. You also have options to create upto 5 VPCs inside a project in case if you need network isolation within your project. Instances in one subnet can talk to the instances in another

subnet using a private IP address as routing is enabled by default between the subnets; however, you can always restrict it by applying firewall rules or network tagging. Note that firewall rules and routes are created in VPC and may have network tags. These network tags are used in GCE VMs, for example, to apply firewall rules and routes.

There are two types of VPC according to the subnet creation mode:

- **Auto mode VPC**: Auto mode VPC is the type of VPC in which one subnet per region is created automatically during the creation time itself. In this mode, there are predefined IP ranges in each subnet, and these predefined IPs do not overlap with each other.
- **Custom mode VPC**: A custom mode VPC is one in which no subnets are automatically gets created at the time of VPC creation. Custom mode VPC gives you the flexibility of creating one or more subnets per region with IP ranges of your choice.

Connectivity options in VPC

The various connectivity options in VPC are discussed as follows:

Shared VPC

If you have a requirement in which resources in multiple projects need to get connected to a single VPC environment, you can make use of shared VPC. All resources connected to a shared VPC can communicate with each other using an internal IP address.

You need to create a host project which will have a shared VPC created inside it. Then you can have a number of service projects which will have access to the host project. Shared VPC allows Org Administrators to delegate resource creation tasks to other projects while making sure that they have centralized control over the network, routing, firewall, and so on. GCP allows you to create multiple host projects, but one service project can get connected to only a single host project. You can create access control policies that can be implemented across all service projects within an organization.

VPC Peering

In case if you want resources in two VPC networks to communicate with each other, you can make use of VPC peering. In VPC peering resources within an organization communicate with each other using an internal IP address, hence, providing reduced latency. There is no change in the administration part when you use VPC peering. Both VPC networks will be administered separately even if VPC peering is enabled. A subnetwork CIDR range in one peered VPC network should be different than the subnetwork CIDR in another peered VPC. Your one VPC network can have VPC peering with multiple VPC networks. In VPC, peering network traffic does not travel externally and stays within the GCP environment. Since resources are communicating with each other using internal IP addresses within the GCP environment, no network egress charges are applicable. A network administrator has the power to create and delete VPC peering between VPC networks. Subnet routing is used to enable communication between the two VPC networks in VPC peering.

GCP Interconnect

In case if you want to connect an On-prem data center to the Google Cloud Platform, then Cloud Interconnect is the option for you. Cloud Interconnect provides a reliable, low latency, and highly available connectivity between your On-prem data center and Google Cloud Platform.

GCP interconnect comes with the following two options:

Dedicated Interconnect

Dedicated Interconnect provides a direct physical connection between your On-Prem data center and Google Cloud Platform. It is a low-latency option available in Google Cloud. A dedicated Interconnect provides you an option to transfer the data upto 200 Gbps using two circuits(2*100Gbps). GCP provides 99.99% availability in the dedicated interconnect option. You need to have a nearby Edge location of the Google Cloud Platform in order to have a Dedicated Interconnect connection. You get an option to select from 10-Gbps single-mode fiber or 100-Gbps multi-mode fiber. Cloud Router is responsible for creating a BGP session with the On-Prem router depicted in the following diagram. Cloud Router gets the routes advertised by the On-Prem router, and these dynamic routes get updated in the associated VPC Network.

Figure 5.4 shows a dedicated interconnect:

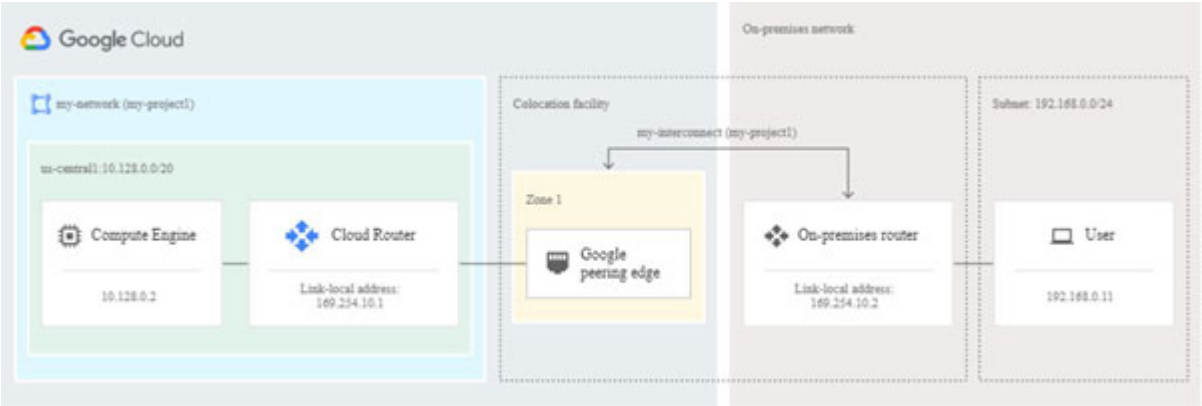

Figure 5.4: *Dedicated interconnect*
Source: *https://cloud.google.com/network-connectivity/docs/interconnect/concepts/dedicated-overview*

Partner Interconnect

When you do not have Google peering edge available at a location that can get directly connected with your On-prem network, you make use of the Partner Interconnect option. Here On-prem environment gets connected to the GCP network through a service provider. Here you have service providers who have direct connectivity with Google's network. When you want to connect your on-Prem data center with Google's network, you connect it with the service provider, post which you can request for a Partner Interconnect connection. Bandwidth supported by Partner Interconnect connection is between 50 Mbps and 10 Gbps and supports 99.9% and 99.99% availability. A VLAN attachment needs to get created in your Google cloud project with your service provider in order to establish Partner Interconnect connectivity. Under Partner Interconnect service provider may ask you to establish either a BGP session between your On-Prem network and service provider or may ask to have a static route configured. You can establish a connection using multiple service providers in order to support high availability.

Figure 5.5 shows a partner interconnect:

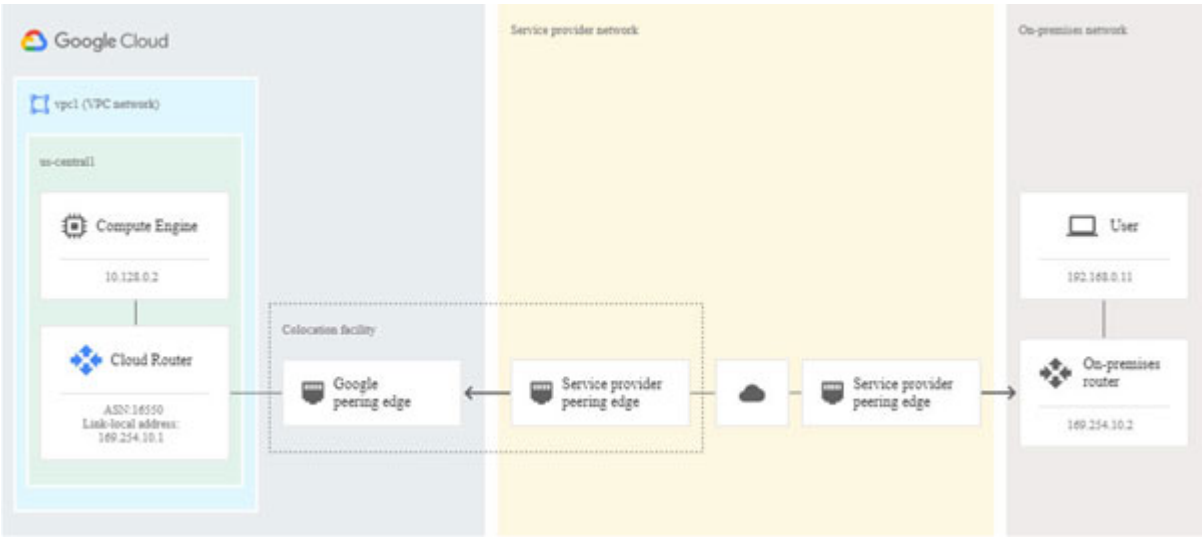

Figure 5.5: *Partner interconnect*
Source: *https://cloud.google.com/network-connectivity/docs/interconnect/concepts/partner-overview*

Connecting to Google Workspace and Google APIs

Peering is another type of connectivity option provided by the Google Cloud Platform, which allows us to connect to Google applications such as YouTube, Gmail, Appsheet, and so on. It allows users to have a connection to access Google Workspace applications. Google Workspace is a new name for GSuite applications.

GCP network peering is of the following two types:

Direct Peering

Direct Peering allows a direct connection between the user's own premise network and Google's Edge Network. It provides high throughput. Google Cloud services are exposed using the public IP address and are directly accessed by customer's On-Prem networks. GCP peering uses a path external to the Google Cloud Platform; therefore, unless there is a need to access Google Workspace applications, it is recommended to use Interconnect. You have to enable direct egress pricing in order to use direct peering. In this connection, end-to-end routes are not propagated; instead, traffic goes from one hop to another hop using the default internet gateway. It can also move using a VPN tunnel.

Carrier Peering

Using Carrier Peering, you can access Google applications through any service providers' infrastructure. Carrier Peering provides you with low latency and high availability, as you can use multiple links for connecting your On-Prem network to the service provider network. Here also, no default routes are propagated, and traffic travels from one hop to another hop using the default internet gateway or VPN tunnel.

VPC creation best practices

There are some best practices that need to be followed while creating a virtual private cloud. Some of the important practices that one should consider while designing their VPC, as described as follows:

- **Use custom mode VPC network**: It is advised to use a custom mode VPC network for multiple reasons. For example, a default mode VPC network uses default subnet IP ranges, which results in IP overlapping between two VPC networks. Therefore, VPC peering is not allowed in default mode VPC network as it may result in IP conflict. Another example is, in case if you want to connect your On-prem network with a default VPC in GCP, it may also result in IP conflict in case on-prem IP matches with the default IP of a subnetwork. A custom mode VPC is also more secure since you only create subnets in regions where you provision resources. It also results in less administration. A custom-mode VPC also helps in meeting compliance requirements.
- **Try using a single shared VPC wherever possible**: Even though users have the option to create multiple host projects, and hence, multiple shared VPCs, it is advisable to create a single host project to create network, security, and IAM policies. Using a single shared VPC within a host project provides ease of management and administration. This best practice is specifically suitable when multiple teams are engaged.
- **Create multiple host** projects if the resource quota limit is exceeding: If the quota limit for a single project exceeds, you create multiple host projects.
- **Create separate VPCs for different environments**: It is advised to create separate VPCs for different environments to enable isolation

and ease of management.

- **Create different host projects for having different IAM control policies**: If you want to implement different IAM policies implemented for different VPC networks, you need to have separate projects created for this requirement, as IAM policies are implemented at the project level.

Please note there are many VPC best practices suggested for VPC. Readers are advised to go through GCP documentation to get more understanding of it.

Cloud NAT

NAT stands for network address translation. NAT is used to convert your private IP to public IPs. If you want to connect your GCP resources to the internet without using an external IP address, you can make use of Cloud NAT service. You can connect your Compute Engine, Kubernetes cluster, App Engine, Cloud Functions, and Cloud Run resources to the internet without exposing them directly to the internet.

In a traditional NAT concept, a proxy NAT service is used in between to convert your private IP to public IP and vice-versa; however, in Cloud NAT and no NAT proxy is used. Here each VM is designed or programmed to use NAT using specific assigned ports. Cloud NAT uses Google's Andromeda software to implement NAT. Cloud NAT is secure, scalable, and highly available that does not consume network bandwidth from the VM. You have the option to assign an IP address manually in order to make sure that only specific external IP addresses are allowed for incoming Cloud NAT at the destination. A specific default static route with a default network gateway is needed in order to connect to a specific route. You can create a single Cloud NAT gateway in order to connect to primary, and alias IP ranges in a resource; alternatively, you always have the option to create additional Cloud NAT Gateways.

Figure 5.6 features Cloud NAT:

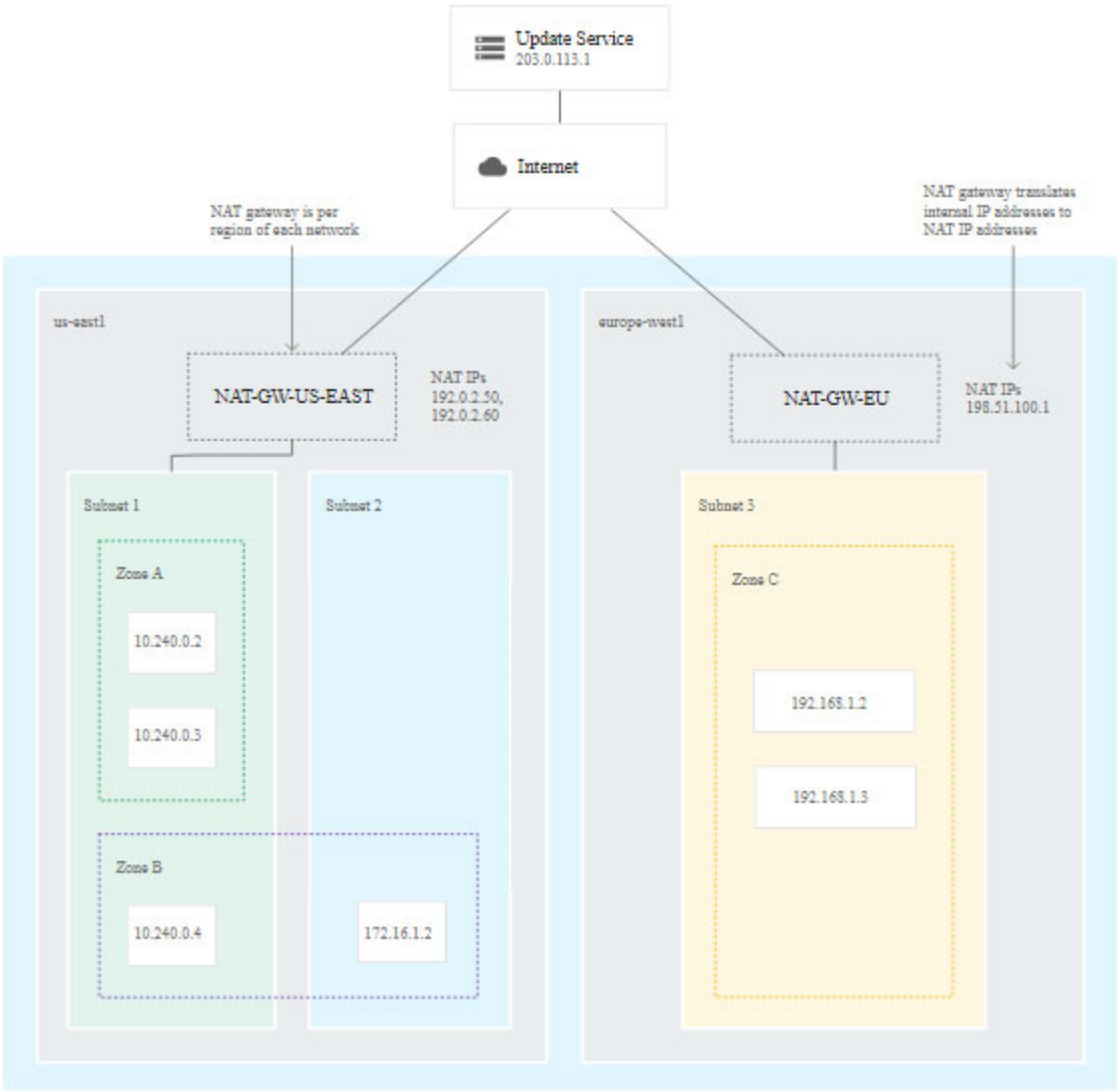

Figure 5.6: *Cloud NAT.*
Source*: https://cloud.google.com/nat/docs/overview*

Cloud Load Balancing

In the present age of digital transformation, where it is common to have unpredictable requests coming in for applications, it is very essential to rely on scaled-up or scaled-out infrastructure.

Scaling up means increasing the CPU, memory, or storage of your compute instance, whereas scaling out means adding an additional number of nodes to an existing cluster. It is challenging to add more CPU and memory to the existing instance as the load is often unpredictable in nature and the

compute instance also has limitations in terms of how much resources can be further added to an instance. Therefore, we have seen that scaling out has been considered as a preferred strategy, in which incoming requests can be processed by some other node if one of the nodes is busy. The load balancer is a service that is required to distribute incoming requests across target nodes with the aim of making optimal utilization of compute resources. Load balancers can either be a Hardware load balancers or Software load balancers. Hardware load balancers require the racking and stacking of physical hardware, which will be used to distribute the incoming traffic to the various nodes, whereas Software load balancers are installed and configured on an operating system.

It is often a time-consuming task to create and manage a load balancer; however, Google Cloud Platform provides Cloud Load Balancer as a service that requires minimum human intervention. Load balancing services in Google cloud can be divided into the following categories.

Global load balancer

Global load balancer helps in distributing the traffic across various geographical regions. As VPC is a global entity, in the case of GCP, there is no VPC peering required before implementing global load balancing. Global load balancers can be used for load balancing, ensuring high availability and disaster recovery. Global Load balancers can be of the following three types:

HTTP/HTTPS load balancer

As the name suggests, this load balancer takes the external http/https traffic that comes in the form of web requests and splits it across back-end services hosted in Google Cloud Platform such as Compute Engine, Google Kubernetes Engine, and so on. It makes use of port number 80/8080/443 to take the incoming traffic and routes it to the back-end services. This Load balancer works on Layer 7, and hence, is capable of doing content-based routing. Traffic first hits the HTTP proxy at the load balancer and redirects the traffic to the respective Managed Instance Group depending upon the policy that has been setup. For example, if one Managed Instance Group in a specific region has reached its max rate or utilization level, then the traffic will be redirected to the other one, which is closest to the region from where

the request has been generated. HTTPS load balancer works as a proxy between your client and the destination application. When an HTTPS request is generated by the client, it is encrypted by default. The load balancer should have a private TLS certificate to decrypt the incoming request before it can be forwarded to the destination.

SSL Proxy Load Balancer

SSL Proxy Load Balancer is used for SSL offloading. It means any kind of decryption of the incoming traffic is taken care of by the load balancer itself; hence, back-end systems are not consuming their compute power to process the decryption activities. SSL Proxy Load Balancer premium tier has the option to load balance the traffic to the respective back-end at different regions. SSL Proxy Load Balancer should be used only for the SSL traffic and not for the HTTPS traffic. A Global Forwarding rule is a setup to ensure the redirection of the traffic to the appropriate back-end.

TCP Proxy Load Balancer

A TCP Load Balancer redirects the TCP traffic coming from the internet to the appropriate back-end. By default, the TCP Proxy Load Balancer redirects the incoming traffic to the closest back-end from where the request is coming. In case if the closest back-end is busy, it redirects the traffic to the nearest region/zone. TCP offloading is done at the load balancer layer itself. In the Standard Tier, the TCP Proxy Load Balancer handles the traffic regionally, whereas, in the premium tier, it is handled globally. It is a reverse proxy load balancer and can use a single IP address for all users.

Regional Load Balancer

As the name suggests, a Regional Load Balancer can help with balancing or splitting the traffic across various nodes within the same region. These nodes can be part of a VM cluster or Kubernetes cluster within the same region. In case if you want to split the traffic across back-end nodes with the same region, you can make use of a Regional Load Balancer. A Regional Load Balancer can be divided into the following categories:

Internal TCP/UDP Load Balancer

If you want to perform regional load balancing on the internal VMs and do not require SSL termination done at the load balancer level, this load balancer can be used for Layer-4 load balancing. Internal TCP/UDP load balancer has a front-end service that has some forwarding Rule defined. It is connected to a back-end service from where traffic gets redirected to a specific instance group in respective zones. Here SSL traffic gets terminated at the back-end.

Network Load Balancer

If you want to load balance external traffic among virtual machine instances within a single region, the Network Load Balancer can be used. The Network Load Balancer supports TCP, UDP, and ICMP protocols. It is a Layer 4 load balancer. The network load balancer can span within a single region. Here, traffic does not get forwarded to a proxy server. Traffic that passes through the Network Load Balancer directly goes to the VMs and can preserve the client's IP address.

Internal HTTPS Load Balancer

Internal HTTPS Load Balancer is a Layer 7 load balancer that load balances or redirects the traffic to VM, instance groups, Google Kubernetes Engine, or Cloud Run within the same region using internal IP. You can use an HTTPS Load Balancer to load balance the traffic or redirect the traffic to a specific endpoint. For example, you can redirect the traffic requesting for audio files to a specific instance group and for movies to another instance group within a region. In a Hybrid environment, an HTTPS Load Balancer can be used to redirect some specific traffic requests to the On-prem environment and others to GCP.

Note: In a GKE cluster, it is recommended to use GKE Ingress Controller for load balancing. GKE Ingress controller automatically deploys a load balancer to load balance the traffic among worker nodes.

Instance groups

An Instance group is a collection of multiple VMs. Using an instance group, you can manage multiple virtual machine instances as a single entity. An instance group in GCP can be divided into two categories.

Managed instance group

A managed instance group is one in which GCP manages the underlying identical VM instances that are part of the group. In this, GCP provides Auto scaling, Load balancing, and Auto healing features with no management overhead. The following are the main features of a managed instance group:

- **Auto-scaling**: Managed instance group can automatically scale the number of instances in a managed instance group. In case if the workload increases, the number of instances will scale out automatically, and when the incoming load reduces, the number of instances will go down. You have to specify the maximum and the minimum number of instances.
- **Auto healing**: Auto healing is another feature provided by managed instance group. In case if an instance stops running due to some reason, managed instance group recreates that instance. In case if any application crashes or hangs due to some reason, an instance will get recreated. A managed instance group uses health check for Auto healing.
- **Load balancing**: A managed instance group load balances the traffic across all VM instances that are part of the same region or zone. You have the option to create a regional Managed Instance group or a Zonal Instance Group.
- **Support for stateful and Stateless applications**: A managed instance group provides support for stateless applications like front-end applications running on Virtual machine instances. Stateful applications such as database applications, Batch computation, and so on can also be managed using managed instance group.
- **Auto updates:** Applications can easily be updated and rolled out to the underlying VMs automatically using managed instance group.
- **Support for containerized workload:** You can deploy container images on the VMs that are part of a managed instance group.

Unmanaged instance Group

An unmanaged instance group is one where unidentical VMs are part of an instance group. The unmanaged instance group does not provide most of the features provided by the managed instance group, which means it does not support Auto scaling, Auto healing, and multi-zone deployment features provided by the managed instance group, and hence, is not recommended for a varying workload. You can load balance the traffic across VM instances that have different configurations.

Note: You do not pay anything extra for an instance group, and you are only charged for the underlying resources that are being consumed.

GKE Ingress Controller

Ingress Object in GKE implements Google Cloud Https Load balancer on GKE. In order to use the GKE Ingress controller, you have to make sure that the Https Load balancer in your GKE cluster remains enabled. Ingress rules define how traffic is going to route in the GKE cluster.

There are two types of GKE Ingress resources:

- **Ingress for external Https load balancer:** As the name suggests, Ingress for Https external load balancer is used to deploy the Global HTTPS load balancer in Google's edge location.
- **Ingress for internal Https load balancer:** Ingress for HTTPS internal load balancer will deploy the Internal HTTPS load balancer using envoy proxy.

Container native load balancing

Container native load balancing is a very important feature provided by the ingress controller. It enables PODs as load balancing targets and load balances the workload across PODs instead of virtual machines. In case if we do not use container-native balancing, traffic gets redirected to the worker nodes first, and from there, it gets redirected to the individual PODs, which may or may not reside in the same worker node. However, with container-native load balancing, traffic gets redirected directly to the PODs

as they become the primary target of load balancing. This results in better distribution of the traffic that provides better load balancing. Container native load balancing is an essential feature of internal ingress controllers; however, for external ingress controllers, you get it as an optional feature. Cloud-native load balancing supports advanced features like Cloud Armor, Identity and Access Protection,, and cloud CDN. We will be explaining about CDN in the later part of this chapter.

Hybrid cloud concepts

Hybrid cloud can be defined as a combination of multiple environments. The most common example of a hybrid cloud is the combination of public and private cloud. There can be various reasons for having a hybrid cloud environment. For example, due to compliance reasons, you do not want to move certain applications out of your data center. In this case, you would like to connect your On-prem data center to the public cloud so that some of your workloads can run on the cloud while others can run on the public cloud. Google Cloud provides various connectivity options to connect your On-prem data center to the Google Cloud Platform. As the requirement of hybrid and multi-cloud environments keeps on growing, so is the need to have a common management platform through which you can manage the applications running on these environments. For managing containerized applications running on On-Prem and Google Cloud environments, GCP has an offering called Anthos, using which you can manage containers running on a multi-cloud or hybrid cloud environment. Anthos offers microservices management using a single pane of glass. It allows users to manage and migrate microservices-based applications. You can troubleshoot the issues related to the interaction between various microservices. Migrate for Anthos in another offering from GCP that helps you with containerizing, modernize and migrate workload. In a hybrid environment, you can always make use of existing DevOps tools and integrate them with cloud-native services that help with the provisioning of infrastructure and creating CI/CD pipelines for application deployments.

Cloud CDN and edge locations

Cloud CDN stands for content delivery network. When you have users across the globe, who are trying to access content residing in some specific

geographical regions, it may result in some latency issues. Google provides various edge locations across the globe that help in connecting clients or users to the Google network. Cloud CDN resides on these edge POPs that are used to cache content that needs to be delivered to the end users.

Figure 5.7 features Cloud CDN:

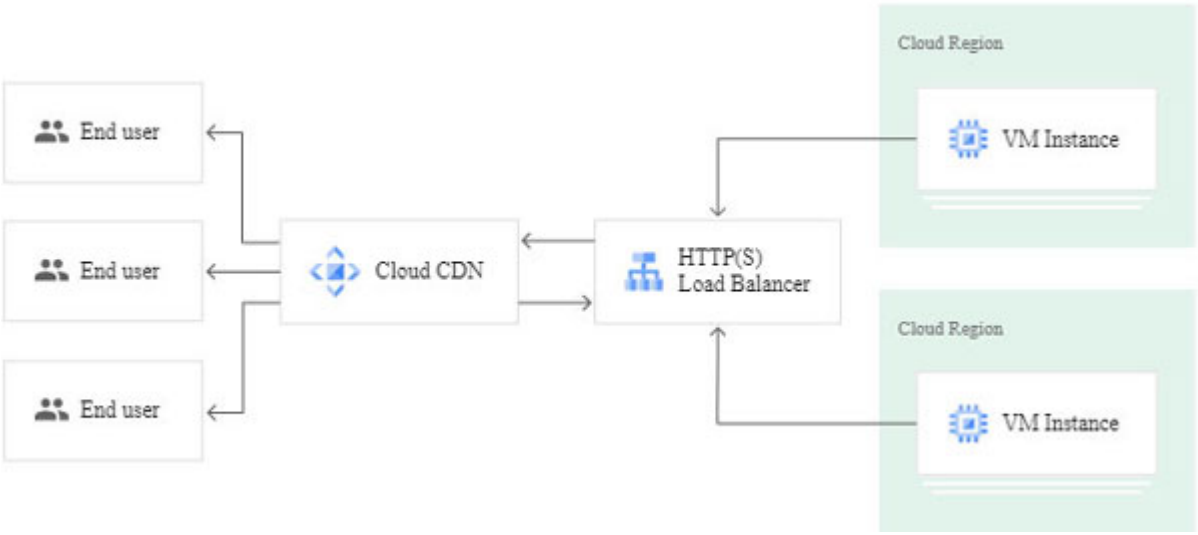

Figure 5.7: *Cloud CDN.* ***Source:*** *https://cloud.google.com/cdn/docs/overview*

Cloud CDN also helps with handling sudden network spikes. In Cloud, CDN content is stored in the form of a cache. The first time when the http request is made to the HTTP/HTTPS load balancer, content is accessed from the actual region where it resides; however, frequently accessed content gets stored in CDN; therefore, when the content is accessed through the Google front end, it finds it in cache, and the response is delivered to the end user. This is called cache hit. Cache miss is the situation when the content is not available in the cache and is taken from the respective region where it resides. You can enable Cloud CDN at the load balancer level. You can also bypass Cloud CDN for some specific content that you do not want to get delivered through Cloud CDN.

Conclusion

In this chapter, we have discussed about various network services and options available in the Google Cloud Platform. We started with understanding the overall organization and resource hierarchy and then explored the VPC network. We also checked various connectivity options

available in GCP for connecting to external or On-prem networks. We understood various load balancer options available in the Google Cloud Platform and discussed briefly about hybrid cloud concept and CDN. In the upcoming chapter, we will be mentioning about various security and monitoring services available in GCP. After going through the chapter, readers will be able to understand the security-related concepts in GCP.

Note: Readers are advised to go through Google's official documentation in order to develop more understanding of this topic.

Key terms

- **VPN**: Virtual private network.
- **NAT**: Network address translation used for private to public IP conversion.
- **Edge locations**: Edge routers are hosted in locations near to the customer. Customers can order a cross-connect to those routers for peering.
- **CDN**: Content delivery network.

Questions

1. Explain the organization structure in the Google cloud platform.
2. What is a Virtual private cloud?
3. What are the various connectivity options available in GCP?
4. What is a load balancer? Explain various types of load balancer options available in GCP?
5. What is a GKE ingress controller?
6. Explain the content delivery network in GCP.

Further reading

- **https://cloud.google.com/products/networking**
- **https://cloud.google.com/vpc/docs/vpc**

CHAPTER 6

Security and Monitoring Services in Google Cloud

Cloud security is a collection of procedures and technology designed to address external and internal threats to business security. Organizations need cloud security as they move toward their digital transformation strategy and incorporate cloud-based tools and services as part of their infrastructure.

-IBM

Introduction

Security has been a primary concern for organizations that are planning to move to public cloud. However, with the ever-growing evolvement of the cloud, various cloud providers have come up with advanced security features just to make sure that the customer's applications and data is well protected from any type of internal and external threats. Google Cloud offers various services just to ensure that the customer's data is well protected while in rest and also during transit.

Structure

In this chapter, we will cover the following topics:

- IAM roles, permissions, and policies
- Service account
- Container security
- Cloud DNS
- Cloud Armor
- Secret Manager

- Key Management Service (KMS)
- DevOps concepts
- Security in DevOps
- Google Cloud Operations Suite
- Security-best practices

Objectives

After going through this chapter, the readers will be able to have a fair understanding of various security and monitoring services offered by the Google Cloud Platform. This will help them in making a decision to select and implement appropriate services for various use cases. Readers are advised to go through GCP's official documentation to develop in depth understanding of these services.

IAM roles, permissions, and policies

Google cloud **identity and access management(IAM)** is one of the most important features of the Google Cloud Platform, as it is responsible for assigning various roles and associated permissions to access various resources in GCP. In order to understand Identity and Access Management and its implementation in GCP, we need to understand the organization and resource hierarchy in GCP. We have already discussed those concepts in the chapter Networking in Google Cloud; however, we would suggest readers to revisit the topic once.

Member

In GCP, a user account, Gmail account, Workspace account, Service account, and a Google group all are an example of a member.

Roles and Permissions

Permissions are a set of activities that you are allowed to perform on resources in GCP. In order to apply for various permissions, we need to have roles. Permissions cannot be assigned directly to users. In order to do that, various roles need to get created that needs to get associated with

members that have various type of users associated with them. Roles are assigned at the organization, folder, projects, and resource level and follow the principle of least privilege. GCP provides the following different types of roles.

Primitive or Basic roles

Primitive roles are the most basic roles in the Google Cloud Platform. GCP provides three primitive roles, namely, Owner, Viewer, and Editor. These roles are assigned at the project or resource level and do not provide granular access control to the resources. Normally, these roles are not assigned due to their lack of granularity. The Viewer role provides read-only permission on resources, whereas the Editor role provides required permission to create or change the resource. The Owner role is used to manage roles and permissions assigned at a project level. The Owner role inherits the Editor and Viewer roles; similarly, the Editor role inherits the Viewer role.

Pre-defined roles

GCP provides a number of pre-defined roles that can be used as it is. It contains the most common roles with permissions that are required for managing the resources. Pre-defined roles are required to provide fine-grained access control on underlying resources. These roles are created and managed by GCP, which means any kind of updates in the pre-define roles are automatically taken care of by GCP. GCP also keeps on adding additional roles as and when required. The only challenge with pre-defined roles is that there are so many roles available that sometimes you end up providing too many pre-defined permissions to a user. Customers get an option to add/remove additional permissions to these pre-defined roles. There are many pre-defined roles that contain unrequited permissions; therefore, one has to be really careful while assigning these pre-defined roles to the users.

Some examples of pre-defined roles are as follows:

- BigQuery Connection User (roles/bigquery.connectionUser)

  Permissions:

  `bigquery.connections.get`

bigquery.connections.getIamPolicy
bigquery.connections.list
bigquery.connections.use

- Storage Admin (roles/storage.admin)

  Permissions:

  firebase.projects.get
  orgpolicy.policy.get
  resourcemanager.projects.get
  resourcemanager.projects.list
  storage.buckets.*
  storage.multipartUploads.*
  storage.objects.*

- Compute Admin (roles/compute.admin)

  compute.*

  resourcemanager.projects.get
  resourcemanager.projects.list
  serviceusage.quotas.get
  serviceusage.services.get
  serviceusage.services.list

Custom roles

As the name suggests, custom roles are the ones that can be created for the specific requirement with specific permissions. When you want to create roles with a specific set of permissions, then you make use of a custom role. You can create custom roles from scratch where you can give them a name and assign specific permissions, or you can make use of existing pre-defined roles and edit them to convert them into custom roles.

IAM policy

We have already discussed about members and roles. IAM policy is created when you bind a specific member to a specific role. It is a way of telling that a specific member is allowed to perform specific tasks on specific resource/resources. You can implement IAM policy at any specific level, i.e., organization, folders, project, or resources. All child objects in an Organization hierarchy inherit the policies of the parent objects. GCP uses

the principle of least privilege. It means that if a member is assigned write access to a resource at the organization level but has only view access at the project or resource level, then the member will have only view access for a specific resource. Whenever a specific member makes a request to access a specific resource, then it is the responsibility of IAM to check if a member is allowed to access that resource or not.

Service account

A service account is a specific type of account that is used by virtual machines and applications to provide access to resources. There are various use cases for a service account. For example, applications running on a VM can make a call to specific Google APIs to perform specific tasks. Another example is any On-prem application making calls to GCP APIs in the Google Cloud Platform. Service accounts have an email associated with them and do not require any password. Authentication is performed using a public/private key pair.

GCP provides two types of service accounts, which are discussed as follows:

User-managed service accounts

As the name suggests, user-managed service accounts are the ones that are managed by the users. If you want some applications to have access to Google cloud resources, you create service accounts and grant them required roles to perform some certain tasks on resources in GCP. User-managed service accounts are created using IAM API and require a password to access the resources that are stored in a service account key.

Google-managed service accounts

Google-managed service accounts are the ones that are created and managed by the Google cloud platform. For example, when you create any resource like Compute Engine in GCP, a default service account for that VM is created. This service account will be used by a virtual machine to access other services in the Google Cloud Platform. Google-managed service accounts do not use any keys to authenticate; instead, use randomly generated tokens to authenticate.

Container security

Containers are prone to many security vulnerabilities. Google Container Analysis does the required vulnerability scanning of container images that reside in the Artifact registries. Whenever any new container images are uploaded in the artifact registry or in the container registry, container analysis performs scanning of the image. It can be either on-demand scanning or automated scanning. You always have options to disable this scanning. Whenever there are any changes in the content, this scanning will be performed. Depending on the results of this vulnerability scanning, one can create an allow list to allow deployment of container images using Cloud Run. You can make use of binary authorization to do that. Restricting access to container images is another way of ensuring that containers remain secure.

Cloud DNS

Domain name system converts domain names into associated IP addresses. It's a hierarchical database that stores information in a more user-friendly way; therefore, users need not to remember IP addresses in order to access any application. Applications can be accessed by the users using a domain name instead of an IP address.

Cloud DNS is a DNS service offered by GCP that is highly available, fault-tolerant, resilient, and globally available. Since this service runs in multiple regions, it makes it highly available and faults tolerant. Cloud DNS provides public zones and private zones. The public zone is available to the internet, whereas the private zone is visible only to the specific VPC. Here you can have an internal load balancer and an internal DNS with the internal domain name. These domain names can be accessed by any backend service. When you have registered a domain name for a website that needs to get accessed by the public internet, you need to create a record in managed zone config that routes the traffic to it. This IP address can be of a load balancer, or it can also be of an individual independent VM in which your application is running. Cloud **Identity and Access Management** (**IAM**) helps in authenticating the users who are allowed to make changes in Cloud DNS. Only users with editor or owner roles are allowed to make changes to cloud DNS.

Cloud DNS forwarding

DNS forwarding is a process of forwarding incoming DNS queries to another DNS server for resolution. In Cloud DNS, you can forward this either to an internal DNS server within the VPC of another project or you can forward it to an external DNS server that is available On-prem or on the internet. For doing this, you can create a forwarding zone in the shared VPC where cloud DNS is configured, or you can create a cloud DNS server policy. You can create inbound/outbound policies in order to enable DNS forwarding for incoming requests/outgoing requests.

DNS Peering

DNS peering allows us to forward any DNS requests to the private zone of a VPC to another VPC in another project. For using DNS peering, we need to create a private zone and configure it to forward requests to another VPC where records are present. DNS peering will be used in cases where one team does not want DNS records to be maintained in a centralized location like shared VPC. DNS peering uses DNS forwarding to forward any request to another project for other service projects who want to maintain a separate DNS record for their applications. Shared VPC will be managing these forwarding rules at the centralized location and forward any request coming for the host project to the associated DNS project for any DNS resolution.

*Figure 6.1*gives an overview of DNS peering in GCP:

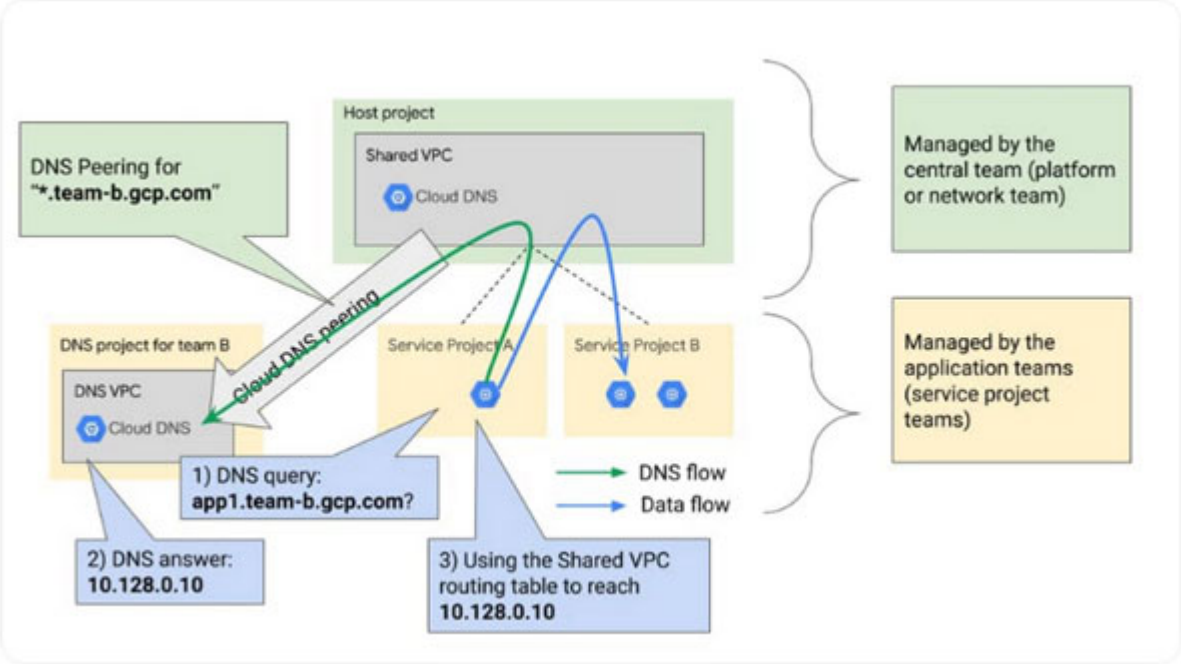

Figure 6.1: *DNS peering in GCP*
Source: *https://cloud.google.com/blog/products/networking/how-to-use-cloud-dns-peering-in-a-shared-vpc-environment*

DNS Security Extension (DNSSEC)

DNSSEC is a feature of the domain name system that prevents it from spoofing. It authenticates the responses received from DNS requests and checks for any manipulation done by attackers in the responses that are generated for those requests.

Cloud Armor

Cloud Armor is a Web application firewall and DDoS prevention service provided by Google Cloud Platform. This service helps users to protect their Web applications and services from external attacks and is available on the Google Edge Network. Cloud Armor works at Layer 7 of the OSI layer. Cloud Armor works alongside GCP's HTTPS Global Load Balancer.

Features

The following are some of the important features of Cloud Armor:

- **Multi-cloud and hybrid cloud support**: Cloud Armor can be deployed on a multi-cloud or hybrid cloud environment. It means even if some of your workloads are running On-prem or on any other cloud, you can still use Cloud Armor to provide the required protection.
- **Rule-based policy**: Cloud Armor provides you with an option to create one or more rule-based policies with security control that can be applied to a specific set of services.
- **Preview option**: Cloud Armor comes with a preview option. As the name suggests, you can always preview the implication of policies in preview mode before they are actually implemented in your environment.
- **Cloud Monitoring and Logging**: Cloud Armor comes with access to the logging option available in the Google Cloud Operations Suite, which helps you in analyzing policies and associated rules.
- **Pre-configured rules**: Cloud Armor comes with pre-configured rules based on the industry's best practices.
- **Adaptive protection**: Adaptive protection is an important feature of Cloud Armor. If Cloud Armor is already implemented in your project, you can enable adaptive protection for the backend services in place. Adaptive protection takes some time to train itself (approximately one hour) to identify the pattern across all services and then create a baseline for the services. Once the training is over, it sends alerts whenever any anomaly is detected in the incoming traffic.
- **Rate limiting**: This feature helps with limiting a large number of incoming requests by blocking them.

In order to use Cloud Armor, you need to create a security policy and then add some rules to it. Post this, you then attach this policy to your service. Cloud Armor security policies can have multiple rules added to them. HTTPS load balancer will check the traffic and see what actions should be taken with respect to some specific rule applied. Cloud Armor also provides a pre-defined set of rules that protects against SQL injections and cross-site scripting.

Secret Manager

Applications running in your environment have many sensitive information associated with them, such as password, encryption keys, IP address, and so on. There is always a need to store this information in a centralized, secure location.

The Secret Manager is a service offered by Google Cloud Platform that is used to store sensitive information such as API Keys, passwords, certificates, and so on. This information is referred as secret. It is a place where you can have centralized control over all the sensitive information.

Features

The following are some of the important features of a Secret Manager:

- **Integration**: Secret Manager can easily get integrated with Application code, Hashicorp Terraform, and GitHub. Apart from this, you can also integrate it with Cloud Identity and Access Manager. This helps you in controlling access to the secret.
- **Replication feature**: Secret names are global; however, they still belong to one specific region. In case if you want to have secrets in another region, then you can specify just the region name, and replication will be handled by GCP at the backend without any manual intervention.
- **Versioning**: You cannot change the secret data. Therefore, every time you have a change, you need to create a new version of it.
- **Audit Logging**: With Audit Logging enabled, you can monitor every interaction happening with secrets. These audit entries are generated by the Secret Manager. These audit entries can be analyzed for any abnormal access pattern.
- **Encryption by default**: Any data stored and in transit is encrypted by default by using TLS and AES-256-bit encryption.
- **Secret management**: Using Secret Manager, you can automate the expiration and rotation of secrets and control their lifecycle.

Key Management Service (KMS)

Key Management Service is used for a cryptographic operation. It stores a cryptographic key inside it and also performs encryption and decryption

using those keys. Key rotation and version control are other features associated with it. KMS does not let anyone access those keys.

The following are the two key management services provided by GCP:

- **Google-managed encryption key**: Google-managed encryption keys are the ones that are provided and managed by GCP. All activities like version control and key rotation is automatically managed by the Key Management Service.
- **Customer-managed encryption key**: Customer-managed encryption keys are created by Google but are managed by the customer. Here key management activities are in the control of the customer. This is used when a customer wants to have some control over managing the keys.

It is always recommended to use Google-managed encryption for its reduced overhead of managing the keys and chances of manual errors with respect to key rotation.

DevOps concepts

DevOps is the combination of tools and technology that is designed to deliver software and other services faster and cost-effective way.

As the name suggests, in DevOps, the development and operations team work together, starting from software development and testing to deployment. Apart from software development and deployment, automatic provisioning of infrastructure resources is also part of DevOps. Some examples of DevOps activities include creating **Continuous integration and continuous delivery** (**CI/CD**) pipelines for continuous integration, development, and deployment with minimal manual intervention using tools such like Jenkins, Ansible, and so on or creating terraform scripts for doing infrastructure provisioning.

Benefits of DevOps

The following are some of the benefits of using DevOps:

- **Speedy delivery**: DevOps helps with faster delivery of applications and speedy provisioning of resources which helps in the completion of

the projects within the desired timeframe.

- **Cost saving**: DevOps practice uses automation for performing various activities. This helps with cost saving as you now require less number of people to perform a set of activities.
- **Better work environment**: DevOps helps in improving collaboration activities between operations and development teams, which results in a better work environment. There are so many activities that are automated, which helps with having lesser conflicts between the team members.
- **Better productivity**: DevOps helps with improving productivity as most of the repetitive activities are automated, and respective teams can focus on their work.
- **Better resource utilization**: Using DevOps practices result in better resource utilization as resources can efficiently focus on the work that they are assigned.
- **Faster problem resolution:** As DevOps practice uses pre-defined steps for performing any task, it results in faster identification of the problem areas, and hence, helps with their resolution.

Some activities under DevOps practice

Some activities that are part of DevOps practice as discussed are as follows:

- **Continuous Integration**: Continuous integration is the practice of merging code developed by various developers in a centralized location like GitHub or any other source code repository.
- **Continuous Delivery**: Continuous delivery ensures that whatever code has been merged to the main branch that is a single source of truth is delivered to Prod, Test, and Dev environment. This can be considered as an automated release process on top of the automated testing of the application code.
- **Continuous Deployment**: Continuous deployment is another step in CI/CD process. It provides an automated way of releasing the application to the production environment as they pass the verification steps.

- **Continuous Testing:** This practice helps with automated and pre-scheduled testing of application or infrastructure code before it is deployed.
- **Infrastructure provisioning using Infrastructure-as-a-Code**: DevOps involves using Infrastructure-as-a-Code for performing automated infrastructure provisioning. In the cloud, using Infrastructure-as-a-Code, we can automate the provisioning of infrastructure resources very quickly and re-use this code to perform provisioning in some other environment as well.

Security in DevOps

DevOps security or DevSecOps can be considered as a set of practices that are followed during the development and deployment life cycle. Before the advent of DevOps, any implementation of security was considered post-application deployment. It was about making sure that the production environment in which the application is being deployed is secure from any external and internal threats and about identifying any architectural issues with the application during security testing. Correcting architectural issues with the application post-security testing can be expensive as the application has already been developed.

In DevSecOps, security starts from the application development stage itself, wherein we try to enable application developers to write the code with security in mind. For example, making sure that the developed code does not have any sensitive information like “username” or “password” as part of the code can be considered as an example of DevSecOps practice. Another example is making sure that the **integrated development environment** (**IDE**) in which code is being developed is secure and has all the necessary security patches are in place.

Implementing DevSecOps

DevSecOps is a culture that works on the principle that security is everyone’s responsibility. Only the implementation of tools and technology is not enough. It is really required to bring that change to people’s mindsets. DevSecOps implementation can be understood by the following points:

- **Cultural change**: As mentioned earlier, getting people to adopt the right mindset is a very important part of DevSecOps implementation. It is very important to educate the employees about the importance of security so that they start adapting to this cultural change.
- **Make info-security an important part of the development and deployment process**: Make sure that your information security team is involved during all steps of application design, development, and deployment life cycle. After completion of every step, there can be a review process for checking any security-specific vulnerability.
- **Get the correct set of tools**: Getting the right set of standard tools for the developers is very important. For example, using a standard set of development tools can help the info-security team to identify any issue in a timely manner.
- **Automated testing:** Developing automated security test cases that are part of the software development and deployment life cycle is very important. Automated testing can be made a part of the build process so that any vulnerability assessment can be done at the initial stage itself.
- **Allocating security budget for the development activities**: It can be very well understood that implementing DevSecOps as part of a cultural change comes with a cost. Therefore, allocating or reserving some security budget for development activities is essential.
- **Make use of monitoring tools**: Monitoring tools play a very important part in DevSecOps implementation. GCPs Google Cloud Operations Suite is one such tool that can be used for identifying any malicious login attempts, unauthorized access, and errors in your application.

Identifying security issues at the early stage of the software development life cycle is of immense benefit, as it detects for any security vulnerability at a very early stage. It reduces the cost to correct these issues. Implementing DevSecOps culture in your organization brings a very positive change as different teams come together and better collaboration results in improving overall quality and productivity.

Google Cloud Operations Suite

Logging and monitoring are an essential part of any IT infrastructure. We have various different tools available in the market, such as Prometheus and Zabbix, which can be used for this purpose. Various cloud service providers have their native tools that are used for monitoring, logging, debugging, and so on. Google Cloud Operations Suite is such an offering from GCP.

Google Cloud Operations Suite is a set of tools that are used for monitoring, logging, alerting, and debugging applications, infrastructure, and services in the Google Cloud Platform. Google Cloud Operations Suite also has the capability of monitoring On-prem and other third-party cloud providers using Ops Agent. It provides detailed metrics and logs using which you can monitor your cloud resources.

The following are the services offered by Google's Cloud Operations Suite.

Cloud Logging

Cloud Logging is a managed service that collects logging data from applications and services and presents it at a single location. Users have the option to customize this location. It means that these logs can be stored in Cloud Storage or can be exported to the data warehouse service BigQuery for analysis. After this logging data is given to BigQuery, users can perform analysis on this data by integrating it with visualization tools like Data Studio.

The following are the main types of logs available in the Google Cloud Platform:

Platform logs

Platform logs are service-specific logs that are used primarily for debugging and troubleshooting issues occurring in Google Cloud services. Platform logs vary as per the services that have been created for the specific organization, projects, or folders.

User-written logs

As the name suggests, user written logs are the logs that are written by a user for custom build applications and services. These logs are exported to

Cloud Logging using Cloud logging API, Logging agent, or Cloud login client libraries.

Security logs

These logs are used to keep a record of any data transaction happening in the Google Cloud Platform.

The following are the various types of security logs available in GCP:

Audit logs

If you want an audit trail for administrative access or data access, it can be found in audit logs.

The following are the various types of audit logs available in GCP:

- **Admin activity logs**: Any kind of admin activity can be monitored using these logs. For example, any API calls or configuration changes to the resources. These logs are enabled by default and cannot be disabled or modified.
- **System event logs**: Any resource configuration changes performed by Google Cloud and not by the end-user or administrator are included in System event logs. These are enabled by default and cannot be disabled or modified.
- **Data access logs**: These logs include logs data when any user reads, writes, or modifies the metadata or resources. These logs are not enabled by default (except BigQuery). It is not recommended to enable these logs unless required as they generate lots of data, and there is a cost associated with it.

Access transparency logs

Access transparency logs are generated whenever any Google employee tried to access your resources. These logs are not enabled by default, and there are specific services in the Google Cloud Platform that generate these logs.

Check the following list to identify the services that generate access transparency logs:

https://cloud.google.com/cloud-provider-access-management/access-transparency/docs/supported-services

Apart from this, Cloud Logging also supports logs generated by any third-party cloud service providers (like AWS) and On-Prem infrastructure. It can be used in a hybrid or multi-cloud environment, and you will have to install Ops Agent in order to enable this logging.

Cloud Monitoring

Cloud Monitoring is used to monitor infrastructure and applications using a visualization tool. It does so by collecting various metrics, event data, and metadata from the Google Cloud Platform, On-Prem infrastructure, or any third-party cloud provider. Cloud Monitoring provides you an option to customize this integration of the semetrics as per your requirement. You can make use of monitoring query language and metrics explorer to analyze metrics and add the required charts to the dashboard. Monitoring CPU, memory, response time, and throughput of the cloud resources are a few examples of Cloud Monitoring. Cloud Monitoring comes with alerting option, which means you can customize an alert whenever a threshold is reached for any metrics or for any uptime check for the resources.

Application Performance Management (APM)

Google Cloud Logging and Monitoring offers the first line of defense against any issues in GCP infrastructure; however, there are scenarios where the problem is not with the infrastructure but with the application itself. Application Performance Management is a set of tools that are used to monitor and debug performance issues in applications. Using APM, you can perform debugging, tracing, and profiling of applications.

Application performance manager consists of the following three tools:

- **Cloud Trace**: Cloud Trace is a tool that is used to analyze any latency data collected via the distributed system. Cloud Trace has the capability to analyze the latency data and generate reports automatically. It can generate near real-time latency data that helps in detecting issues with any application for any recent change. Cloud Trace is very helpful in identifying the amount of time that an

application takes to handle any request. Cloud Trace is easy to setup and analyzes the performance of applications running on VM, containers, and App Engines. Cloud Trace can visualize the Traces across multiple projects within a single window.

- **Cloud Debugger**: Cloud Debugger is another important tool that is used to debug the application running on GCP without accessing application code. Cloud Debugger does not stop the application or add any significant latency to it. It adds less than 10 ms latency at the time of capturing the application state. Apart from supporting Google Cloud native source code repositories like Google container registry and Artifact registry, it also supports other third-party repositories such as Bitbucket, GitHub, or GitLab. Google cloud debugger supports Python, Node.js, Java, Go, Ruby, PHP, or .NET Core programming languages. It is very helpful in situations where you cannot debug the application code locally as the issue is arising in the production environment itself. You can take a debug snapshot after specifying any specific location in the application code. Because of this, the performance of your application does not get impacted.
- **Cloud Profiler:** Cloud Profiler helps you in identifying the specific area of an application code that consumes more CPU and memory. In this way, it helps in identifying poorly written application code that consumes more compute power. This application can be running on a VM, Containers, App engine, Dataflow, or Dataproc. Cloud Profiler does the profiling depending upon the language in which the application code is written.

Table 6.1 is the list of profile types and supported languages:

Profile type	Go	Java	Node.js	Python
CPU time	Y	Y		Y
Heap*	Y	Y	Y	
Allocated heap*	Y			
Contention*	Y			
Threads*	Y			
Wall time*		Y	Y	Y

Table 6.1: *Profiling*

***Please look at the Key terms at the end of the chapter for definitions.**

Different services in the Google Cloud Platform have some restrictions with respect to the languages that are supported. *Table 6.2* tries to depict the same:

Environments	Go	Java	Node.js	Python
Compute Engine	Y	Y	Y	Y
Google Kubernetes Engine	Y	Y	Y	Y
App Engine flexible environment	Y	Y	Y	Y
App Engine standard environment	Y	Y	Y	Y
Dataproc		Y		
Dataflow		Y		Y
Outside of Google Cloud	Y	Y	Y	Y

Table 6.2: *Languages supported for various services*

Table 6.3 shows Operating systems and corresponding Languages supported in profiling.

Operating systems	Go	Java	Node.js	Python
Linux glibc implementation of the standard C library	Y	Y	Y	Y
Linux must implementation of the standard C library	Y	Y (Alpha)	Y	Y (Alpha)

Table 6.3: *Supported operating system*

Profiling agent collects the data, which is shown in the console interface. The console interface helps you in analyzing it.

Figure 6.2 shows how the profiler looks like in the GCP console once profiling has been enabled:

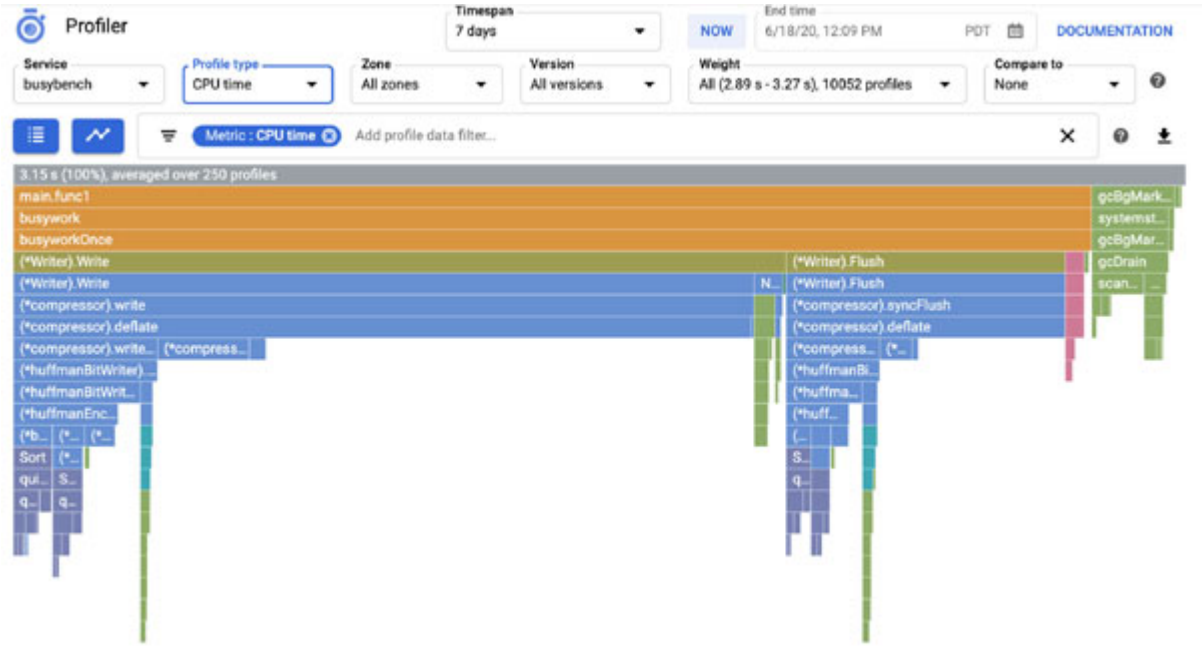

Figure 6.2: *Profiler in GCP console*
Source: *https://cloud.google.com/profiler/docs/about-profiler*

Security-best practices

Security-best practice is a big topic, and it will be impossible for us to cover the entire topic, however, considering the scope of this book, we will try to cover the most important points.

Following are some pointers that can be taken into consideration from the cloud security point of view:

- **Access only to domain users**: While assigning user permissions, make sure that any user who is not part of the organizational domain is not granted access. By default, you are not allowed to add any e-mail account outside the organization, but this feature can be enabled. You can use domain-restricted sharing to prevent this. Also, try to make use of multi-factor authentication for additional security.
- **Cloud Storage**: We need to make sure that the cloud storage is not publicly accessible as it will allow anyone to access the cloud storage.
- **Key Management Service**: We have already discussed about **Key Management Service(KMS)** in this chapter. Users are suggested to make use of Key Management Service as it manages the key at a

secure place. Also, it is suggested to use Google managed key instead of user-managed key, as user managed key is susceptible to unauthorized access. Google-managed keys are rotated frequently and are not accessed by the users.

- **Separate VPC subnets**: Create separate VPC subnets for different environments. For example, Prod, Dev, and Test environment should be isolated and kept in separate subnets.
- **Avoid using public IP for internet-facing application**: It is recommended not to use public IP even at the web tier for internet-facing applications, as it exposes the application directly to an external thread. Instead, try using a bastion host in front of the application and assign public IP to it. You can also make use of an Identity Aware Proxy.
- **Use custom VPC as much as possible**: It is recommended to use custom VPC instead of using default VPC as with default VPC, there are always chances of IP overlapping in case if there is a peering requirement.
- **Use of shared VPC for networking and firewall configuration**: It is recommended to make use of shared VPC for configuring networking and firewall rules in a service project and have one more host projects access to it. It helps with centralized management of networking and firewall, resulting in less chances of missing out on anything.
- **Serial port connectivity**: Ensure that serial port connectivity is disabled by default, as it provides system-level access. As IP whitelisting for the serial port is not enabled, there are chances that the client can access the serial port using its IP and have access to sensitive information such as username, SSH Keys, and so on.
- **Implement the principle of least privilege**: The principle of least privilege states that assign a minimum level of permission to execute a task. For example, a user should not be assigned an admin role on a resource if he just needs to view it.
- **Role assignment**: It is always recommended to assign a role to a group instead of any individual user, as it makes it difficult to manage. You may forget to remove the user from a specific role, due to which he may still have permission to access or edit a resource.

- **Use of custom service account**: Service accounts are a great way to provide application or VM-level access to a GCP resource; however, if, for some reason, application credentials are compromised, the hacker will have access to the resources with the same set of permissions. The default service account has an editor role assigned to it; hence, it is suggested to use a custom service account with the correct set of roles.
- **GKE security**: There are various security-specific recommendations that are given for a GKE cluster. It is suggested to encrypt the cluster node with a customer-managed key. For this, you can use a Key Management Service. It is also suggested to have application-level encryption enabled. This provides an extra layer of security. Apart from this, one can also consider using the authorized network to restrict access to control plane endpoints using HTTPS. This will make sure that only authorized networks are granted access to the GKE cluster.
- **Cloud Logging and Monitoring**: Make sure to configure the audit policy correctly so that any kind of unauthorized access to the resources can be tracked. Admin activity logs are enabled by default and may require to get configured properly. Data access logs can be enabled depending upon the type of application that you are running, as it comes up with some additional cost.

There are just a few GCP cloud security practices. Readers are advised to go through the official Google documentation to develop a further understanding of this topic.

Suggested Link: **https://cloud.google.com/security/best-practices**

Conclusion

In this chapter, we have discussed various aspects of Google Cloud Security and the best practices. Cloud Security is a very detailed topic and should be considered more like an essential practice for any cloud or On-prem implementation. Organizations are in constant need of security professionals who have expertise in this area. It is a very demanding field with immense opportunities. Readers are advised to go through online documentation and videos to enhance their knowledge.

In the upcoming chapter, we will discuss big data concepts and various services available in GCP. We will discuss ETL/ELT concepts in detail and will also develop an understanding of the data warehouse service available in the Google cloud platform.

Key terms

- **WAF:** Web Application Firewall
- **KMS:** Key Management Service
- **ETL/ELT:** Extract transform load/ Extract load transform
- **DevSecOps:** Development security operations

Questions

1. What is Cloud Security?
2. Explain IAM roles, permissions, and policies in detail.
3. What is WAF service available in GCP, and how does it work?
4. What is KMS? What is the difference between Customer-managed encryption keys and Google-managed encryption keys?
5. What is the difference between a Secret Manager and a Key Management Service?
6. What is DevSecOps, and why it is used?
7. Explain security best practices that can be used in GCP.

Further reading

- **https://cloud.google.com/security**

CHAPTER 7

Big Data in Google Cloud

"Big data" is high-volume, velocity, and variety information assets that demand cost-effective, innovative forms of information processing for enhanced insight and decision making."

- Gartner

Introduction

Data is considered one of the most important assets for any organization. Although any unprocessed raw data cannot give us meaningful insights, once processed and transformed, it can be of tremendous use. This data can have a variety of usages. For example, it can be used for analytics that can help in making any futuristic decisions or can also be used in machine learning for training machine learning models. There have been tools and technologies in place that have helped us in getting meaningful insights from raw data after performing the required transformation. As the use of the internet and mobile phones grew, every end user started generating a large amount of structured and unstructured data through social media sites, e-commerce sites, and so on. Different business owners also started understanding the importance of this data as it could help them understand end users' interests, which would help them in coming up with the correct business strategies.

Structure

In this chapter, we will cover the following topics:

- Big Data
- ETL/ELT concepts
- Cloud Pub/Sub
- Dataflow
- Dataproc

- Data Fusion
- Data analytics
- Data visualization tools in GCP
- Google Cloud Cortex Framework
- Reference architecture showing multiple data services in GCP

Objectives

After studying this unit, the readers will be familiarized with the various Big Data concepts and services that are used within GCP to perform various tasks that help to bring data to the desired state. As we understand, data can come in various forms. It can be structured, semi-structured, unstructured, streaming data or batch data, which may be required to get transformed in order to become usable for different requirements. After studying this unit, you should be able to develop some understanding of the previous concepts and also about various Data services available in the Google Cloud Platform. As Big Data is a very big topic, readers are advised to go through various blogs and documentation to develop more understanding of it.

Big Data

Big Data is a large amount of diverse data that is growing at a massive scale. This data can come from various sources, that is, data generated within the organization, through social media sites, government agencies, and so on. Information technology is touching each and every part of our life. Due to the rapid growth of the internet, every section of society has become part of this IT boom. While, on the one hand, devices like smartphones have helped person staying in remote location to stay connected with other parts of the world, it has also brought the challenge of managing the massive amount of data that is being generated at an exponential rate. IoT enabled devices, social media sites, and e-portals are a few examples of internet-enabled services that are generating data that needs to get managed so that it can be used for various purposes.

Structured versus Unstructured data

Structured data is the kind of data that follows a fixed schema or data model and is highly organized. Some examples of structured data are employee

information stored through an organization's e-portal or citizen's data stored by government agencies. Structured data usually have some numeric values associated with it that help in managing it. Some examples of structured data are name, age, education qualification, gender, and so on.

Unstructured data is the type of data that is not stored in a specific predefined schema. This type of data is often generated through social media sites, IoT devices, media or entertainment sites, audio-video data, and so on. Unstructured data may be generated in high volume and velocity, hence, requiring different ways of managing it. At present times, this unstructured data has tremendous business value, as it helps in identifying various trends. There is always a need for analyzing this data to identify end users' interests.

Streaming data versus batch data

Streaming data is the data that is being generated continuously and usually has a continuous processing requirement. This processing is done in real-time. Some examples of streaming data are log data, IoT data, and so on. This data usually have instant analysis requirement and can be used for various purposes such as fraud detection, sentiment analysis, log analysis, and so on.

Batch data is the data that gets collected over a period of time and does not have an instant processing requirement. Once collected, this data can be used for analysis. Sometimes batch data is generated through a legacy system as that system is not capable of generating streaming data. Payroll data and billing data are some examples of batch data.

Data processing tools

As data generation continues to grow at a rapid rate, the need for processing them faster and effectively continues to grow as well. There is always a need to transform unstructured data in order to use it for various purposes. There are various popular tools available in the market that are used for processing this data. Here, we will try to touch base with a few tools in order to develop our understanding of various GCP offerings.

- **Apache Hadoop**: Apache Hadoop is a popular open-source framework that is used to store and process a large amount of data. It uses multiple systems to process a large amount of data in parallel. Hadoop uses **Hadoop distributed file system** (**HDFS**), which provides high throughput and MapReduce operation that helps in processing the

datasets in parallel. The Hadoop Cluster contains two types of Nodes—NameNode and DataNode. NameNode manages the file system configuration and is responsible for distributing chunks of data across various Data Nodes. DataNode is responsible for the actual processing of this data. Map task takes input data and converts this data into datasets, which get distributed across Data Nodes for processing. Reduced task takes the output, performs the required aggregation, and generates the desired result. The Hadoop ecosystem contains various tools and applications. Apache spark is one such system that is used for performing distributed in-memory batch and stream processing. *Figure 7.1* represents the MapReduce function:

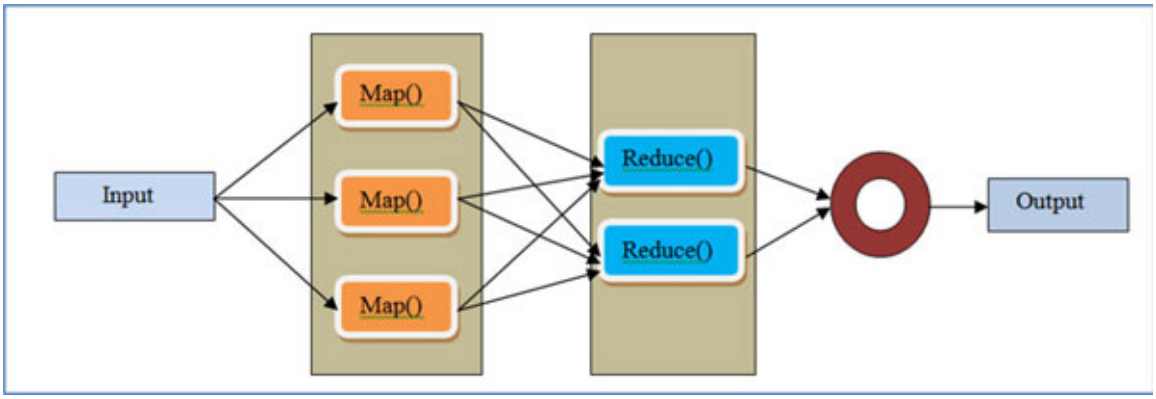

Figure 7.1: *MapReduce*

- **Apache Beam**: Apache Beam is a programming model that is used to define data pipelines used for batch and stream processing. Apache Beam is open source. A data pipeline is a set of actions or steps that are required to transform and move data from source to destination. We will be discussing about data pipelines in detail in the upcoming section on ETL/ELT concepts. Apache Beam supports Java, Python, and Go using which you can build data pipelines. Apache Beam provides SDKs to build these pipelines.
- **Apache Spark**: Apache Spark is an analytics engine that is used for large-scale data processing requirements. Apache Spark does parallel processing in a cluster of compute systems and also uses some set of libraries such as Spark SQL, MLlib, GraphX, and so on. It takes input streaming data, divides this data into batches, and after processing, generates the stream of results. This output data can be consumed by various target databases, data warehouses, dashboards, and so on. The difference between Hadoop and spark is that spark processes the data in

the memory, whereas Hadoop reads/writes data to Hadoop distributed file system.

- **Apache Kafka**: Apache Kafka is a distributed messaging system. It is a messaging system that enables the communication between two applications while it transfers streaming data from one application to another application. It is also defined as a distributed event store and streaming platform that supports both queues and pub-sub models. In a pub-sub messaging system, there is one or more publishers and one or more subscribers. The publisher publishes the message on a topic, and subscribers subscribe to the topic to receive the message. Apache Kafka uses asynchronous communication between the services. It means it just sends the message and forgets about it, which results in avoiding any latency issues. Its primary use cases would be in real-time data feed scenarios, data pipelines, and so on. *Figure 7.2* depicts a high-level overview of Apache Kafka:

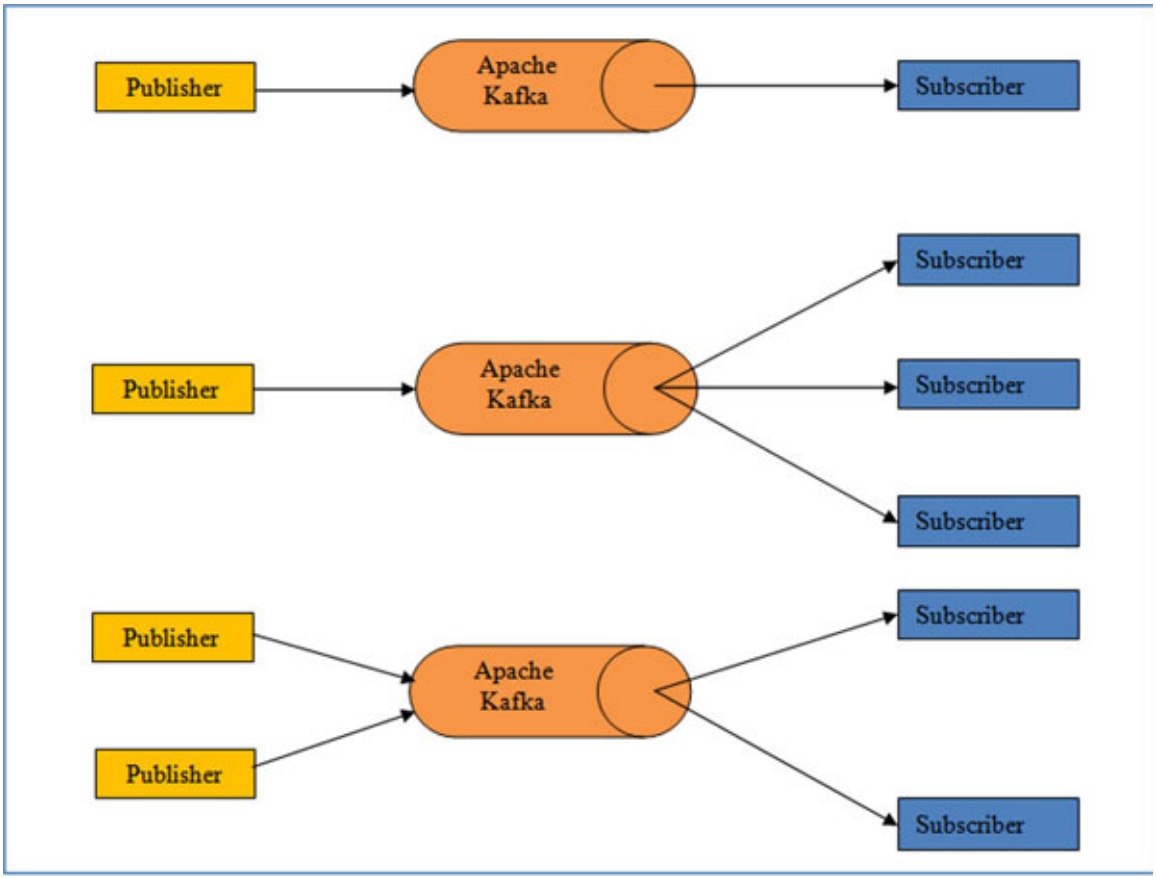

Figure 7.2: *Apache Kafka*

ETL/ELT concepts

ETL and ELT are two very important concepts in data science. ETL stands for extract, transform, load, whereas ELT stands for extract, load, and transform. Data comes from a variety of sources in different formats. Sometimes it can be structured data that is coming from a database, and sometimes, it can be unstructured streaming data that is being generated from IoT devices. In order to make this unstructured data understandable, we may require to process this data and put it into some service like a data warehouse. Once this processed data is available in a data warehouse service link BigQuery, we can run SQL queries against this and get some meaningful insights. Extraction is the process of extracting the data from these sources and ingesting this raw data in a staging area. Once this data is in the staging area, you can perform required transformation on this data before loading it to a target system like a data warehouse or Cloud Storage. The main difference between ELT and ETL is that in ETL, data is transformed first before it is loaded into the target system, whereas, in ELT, data is loaded into a target system before performing any transformational activities. ELT is primarily used in scenarios where we need faster implementation of data movement. However, data loaded into the target system is not clean. ETL is a cleaner approach and transforms the data into the desired state before it is loaded into the target system. *Figure 7.3* depicts the differences between ETL and ELT:

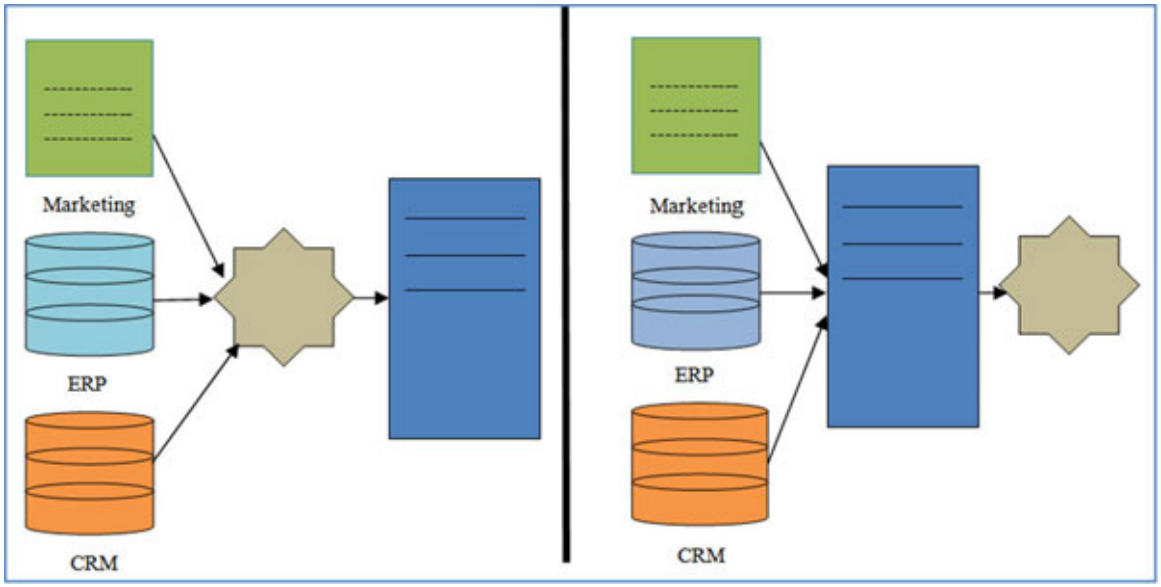

Figure 7.3: *ETL versus ELT*

Cloud Pub/Sub

Cloud Pub/Sub is a messaging service offered by Google Cloud Platform that is used to exchange data between applications and services. Three main components of Cloud Pub/Sub are publisher, subscriber, and topic. Publisher application writes data to a topic. Other applications or services can subscribe to these topics. Pub/Sub allows you to create data pipelines and is used for ingesting the data coming from various sources. As Pub/Sub is an asynchronous service, it does not wait for a confirmation from subscribers after they have received the data. Cloud Pub/Sub is a global service and is available across all regions. Pub/Sub clients are not aware of the location from where they are accessing the data. *Figure 7.4* is a diagrammatic representation of Pub/Sub:

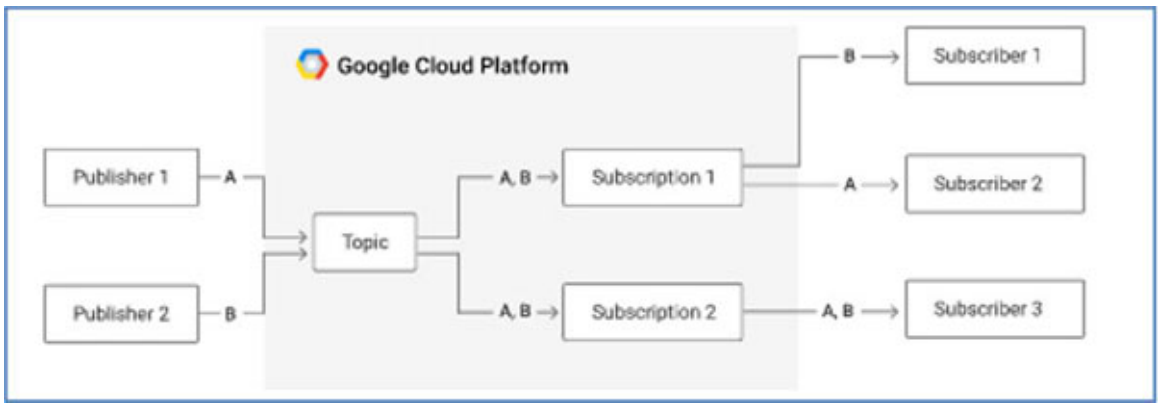

***Figure 7.4**: Cloud Pub/Sub.*
***Source:** https://cloud.google.com/pubsub/architecture*

Important concepts in Cloud Pub/Sub

- **Publisher**: Publisher is an application that sends a message to a Topic.
- **Topic**: The Topic is a resource to which the Publisher application sends a message.
- **Subscriber**: Receives messages on a specified subscription. Messages are sent to topics, and a subscriber may receive them through a subscription.
- **Message**: A message is data that is sent by the publisher to a topic. This message is finally received by the subscriber.
- **Push**: A delivery method where Pub/Sub pushes the data or a message to the subscriber service or application.

- **Pull**: A delivery method in which the subscriber application or service pulls the message or data from Pub/Sub service.
- **Acknowledgment**: A message sent by a subscriber indicating that the message has been received.

Important use cases in Pub/Sub

- **Event notification**: It has always been a challenge to send event notifications in parallel to a number of subscribers. We cannot expect subscriber service or application to wait indefinitely for an event to occur. Pub/Sub helps us in overcoming this problem by sending event notifications to the subscriber services or applications as soon as it occurs.
- **Data replication between databases**: Cloud Pub/Sub can be used for notifying any change event in the source database so that changed data can be replicated to the target database.
- **Log distribution:** Another important use case of Cloud Pub/Sub is that it can be used to distribute the logs across multiple destinations in parallel. You can also use Pub/Sub to send the same logs to multiple subscribers.
- **Real-time data streaming:** Data streaming is a very important use case of Cloud Pub/Sub. Streaming data can get generated from various sources, such as IoT devices and Video Streaming sources. Cloud Pub/Sub is an important component in ingesting this data to various cloud products through a data pipeline.

There are various other use cases available for Cloud Pub/Sun. Readers are advised to go through Google's official documentation to develop more understanding of this.

Dataflow

Incoming data in the data pipeline can be in various formats. It can either be batch data or streaming data. This incoming data is not always usable for destination systems or services. In order to make this data usable to the downstream systems, we need to transform it. GCP provides Dataflow as a scalable, serverless service that can transform this incoming stream and batch data so that it becomes usable for other purposes like analytics or creating machine learning models. It provides data transformation jobs that are written

using Apache Beam libraries. It allows users to create these jobs or use the readymade templates by removing the additional overhead of managing the underlying infrastructure.

When you create a pipeline, you use the concept of Pcollection. Incoming data from various sources are read in Pcollection. Pcollection works in parallel across various systems for parallel processing. Once the processing is done and the data is finally transformed, data is written to the final Pcollection from where it sinks to the upstream system. Dataflow uses Worker Nodes to run jobs. These Worker Nodes can scale out automatically as and when the workload increases. *Figure 7.5* depicts Pcollection:

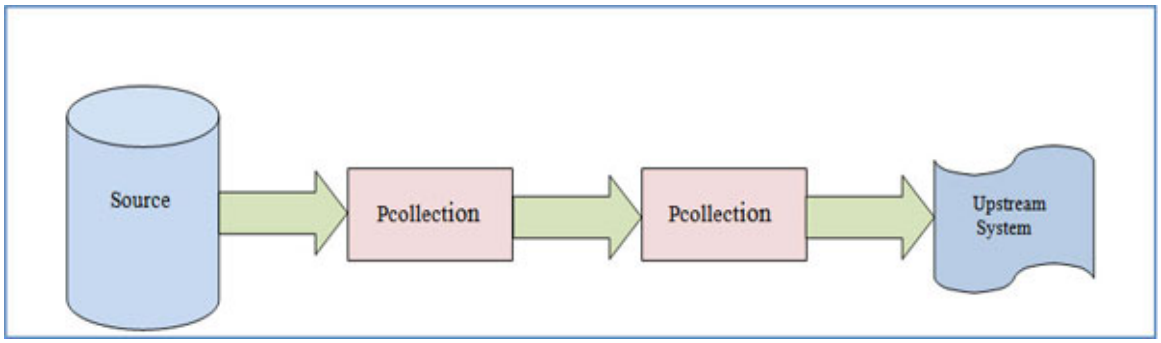

***Figure 7.5**: Pcollection*

Important features of Cloud Dataflow

- **Autoscaling**: Horizontal and vertical autoscaling are two important features of Dataflow. Compute capacity of worker nodes in a Dataflow automatically increases as the load increases. Dataflow can also automatically select a number of worker nodes according to the jobs. It can also scale in and scale out.
- **Dataflow templates**: Dataflow provides readymade pipeline templates that can be used to perform data processing activities. It also provides an option to create templates that can be reused by multiple users for their transformation activities.
- **Inline monitoring**: Dataflow provides an option to do live monitoring of pipelines using charts. You can also create alerts for any latency issues.
- **Dataflow SQL**: Using Dataflow SQL, you can create a streaming pipeline directly from the BigQuery user interface, which is a data warehouse service offered by GCP.

- **Dataflow shuffle**: Dataflow shuffle is an important feature of Cloud Dataflow. A shuffle is an activity that is used for moving and joining the data. Dataflow shuffle shifts these activities from worker VMs to the service backend for batch processing.
- **Flexible resource scheduling**: Dataflow enables flexible resource scheduling by making use of a combination of pre-emptible VMs, regular VMs, and Dataflow shuffle features.

Figure 7.6 is a typical Dataflow example. Streaming data coming from IoT devices are supposed to be used for analytics. Data is first ingested into Cloud Pub/Sub messaging service. Cloud Pub/Sub sends this data to Cloud Dataflow, where this data is transformed and pushed to BigQuery. Here, BigQuery can be used for performing required analytics on this data. We will be learning more in detail about Cloud BigQuery in the upcoming topic.

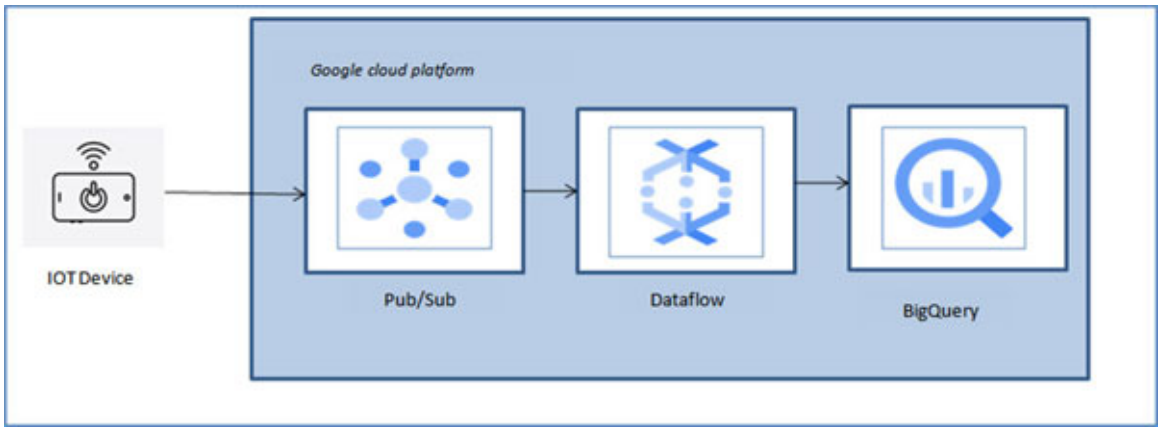

Figure 7.6: *A typical Dataflow example*

Dataproc

Dataproc is another important managed service offered by GCP, which is used for running Hadoop and Spark jobs. Dataproc makes use of compute engine instances for running data processing jobs but takes away management overhead from the users. You do not need to create or manage a cluster by yourself, as Dataproc takes care of it. You just need to run a job for transforming any incoming batch data. It provides easy integration with the other Google Cloud services such as BigQuery, Cloud Storage, and so on and does faster processing of data transformation jobs. Dataproc uses ephemeral clusters and the right amount of CPU and memory that helps in saving the cost.

Benefits of Cloud Dataproc

Following are some of the benefits associated with Dataproc

- **Integrated service**: Dataproc has built-in integration with GCP services such as BigQuery, Cloud Storage, Cloud BigTable, and so on, so you need not explicitly integrate Dataproc with these services while creating your ETL pipelines.
- **Low cost**: Cloud Dataproc uses ephemeral clusters that can scale in and scale out as per the requirement for running a job. It also uses preemptible instances that run as and when the need arises. Apart from it, Dataproc is cheap, as it costs you 1 cent per virtual CPU. This help with bringing down the cost as you are only paying for the time when your jobs run.
- **Managed service**: Dataproc is a managed service. There is no need of configuring the Dataproc cluster for running the jobs. Users can turn off the cluster once the job is finished.
- **Fast**: It requires time to bring up a new Hadoop or Spark cluster on your On-prem environment. With Dataproc, it takes very less time to start and stop or shutdown your cluster.

Important features of Cloud Dataproc

Following are some of the important features of Cloud Dataproc:

- **High availability**: Dataproc allows you to run jobs in multiple nodes, and if one node goes down, the job can get restarted in another Worker Node.
- **Autoscaling**: Cloud Dataproc provides autoscaling. So, nodes can be added and removed dynamically as the load increases or decreases.
- **Cluster configuration**: Dataproc gives you the option to change the configuration of the nodes. You get options to select the machine type, node count, disk size, and so on.
- **Image versioning**: Dataproc allows you to choose from various versions of images for Hadoop, Spark, and so on.
- **Use of templates**: Like Dataflow, Dataproc also gives you options to choose from different reusable workflow templates. Workflow templates

define workflow configuration and area flexible and easy-to-use way of executing workflows.

- **Auto and manual configuration**: Even though Dataproc automatically does hardware and software configuration for you, you still have some manual control to change this configuration.

Figure 7.7 is a Dataproc example or use case. In this, Dataproc uses the readymade connector or template to read the data from a BigQuery, convert it into an AVRO file, and then store it in Cloud Storage Bucket. Dataproc uses a BigQuery connector in order to connect to BigQuery.

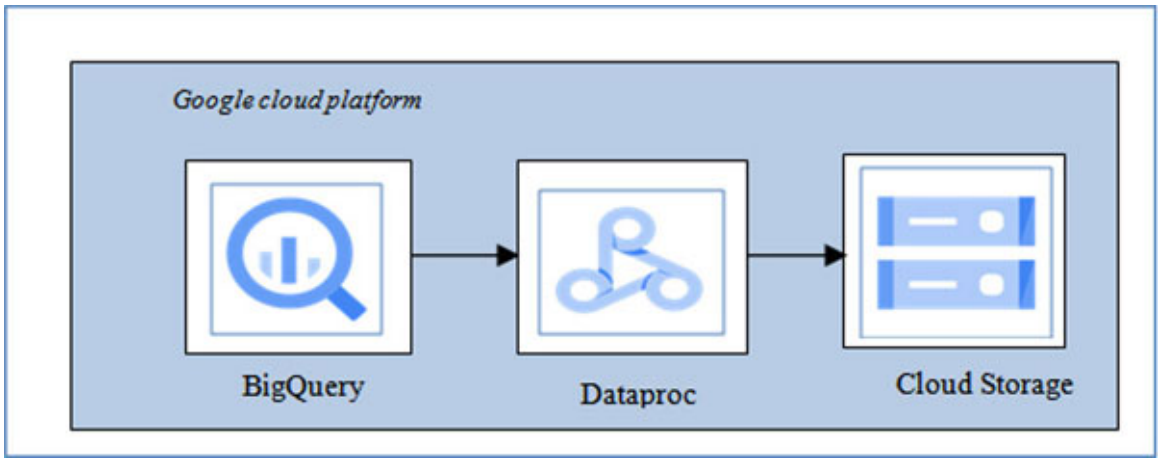

Figure 7.7: *A typical Dataproc example*

Cloud Data Fusion

Data Fusion is another fully managed service provided by GCP that is used to create integrated ETL or ELT pipelines using GUI. Data Fusion takes input batch and streaming data from various sources residing in On-Prem or cloud environments transforms it, and loads it into the target system. No coding skills are required to perform these activities as Dataflow uses readymade connectors and transformers that can be used with the click of your mouse. Data Fusion is built using CDAP, which is an open-source framework for developing analytics applications. Data Fusion is recommended in scenarios where a high level of customization is not required while building transformation pipelines, and you do not have a team having coding skills. Cloud Data Fusion also allows to build pipelines using CLI.

Important features of Cloud Data Fusion

Following are some of the important features of Data Fusion:

- **GUI-enabled service**: Cloud Data Fusion provides code-free GUI to perform transformation service. It provides readymade connectors and transformers to create ETL pipelines.
- **Option for creating custom connectors**: Cloud Data Fusion also provides internal APIs to build customer connectors and transformers that can be shared with others.
- **Batch and Real-time data integration**: Data Fusion supports batch and real-time integration with various data sources. You can replicate transactional and real-time databases such as SQL, Oracle, and MySQL into GCP native data warehouse services like BigQuery. It also allows you to build data pipelines from various relational, non-relation databases.
- **Readymade connectors for various data sources**: Data Fusion provides readymade connectors for integrating with various databases, mainframe systems, and SAP systems that help with building data pipelines.
- **Support for Hybrid and multi-cloud environment**: Data Fusion supports data sources from various sources like On-Prem systems and other cloud service providers.

Benefits of using Cloud Data Fusion

Following are some of the important benefits of using Data Fusion:

- **Fully Managed service**: GCP Data Fusion is a fully managed service. Hence users need not to bother about the underlying complexity. This allows them to focus on building transformation pipelines.
- **Increased productivity**: Data Fusion uses drag-and-drop features to create pipelines. Due to this, you do not need coding knowledge. It helps in saving time for building transformational pipelines, and hence, increases overall productivity.
- **Reduced TCO**: As it is a serverless offering, you do not need people to manage the underlying compute resources. This helps in reducing the total cost of ownership.

Data Fusion comes with three editions, namely, Developer, Basic, and Enterprise. These options vary as per the number of simultaneous pipelines and the number of users supported.

Please check the following link to understand the various features and pricing model support by Data Fusion: **https://cloud.google.com/data-fusion**

Important plug-ins used in Data Fusion

Following is the list of some important plug-ins that are used in Data Fusion:

- **Cloud Storage source plug-in**: This plug-in reads the source data from the Cloud Storage bucker for performing transformational activities.
- **Cloud Storage sink plug-in**: This plug-in is used to sink the data to cloud storage after performing necessary processing.
- **BigQuery**: This plug-in is used to read BigQuery content.
- **BigQuery table or BigQuery multi-table**: This plug-in is used to sink and write the data into the BigQuery table. Data is first written to a Cloud Storage bucket before it is written to BigQuery tables.
- **Oracle**: This is used to read data from the Oracle database.
- **SAP operational data processing**: This plug-in is used for reading the data from the SAP source through operational data processing.
- **Salesforce**: This plug-in is used for reading objects from the Salesforce batch.
- **SQL Server**: This plug-in is used for reading the data from a SQL server.

Figure 7.8 depicts a typical use case of Data Fusion, where Data Fusion helps with ingesting the data from various data sources, processes it, and then sends it to BigQuery and visualization tools for analysis:

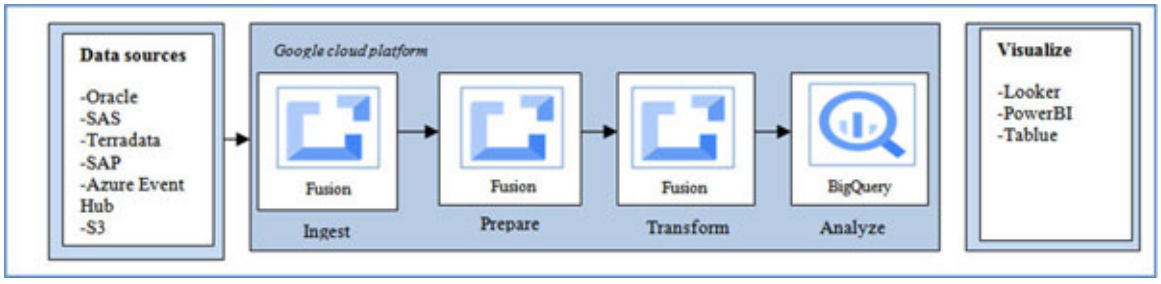

Figure 7.8: *Data Fusion example*

Data analytics

Data analytics is the process of analyzing the raw data in order to identify certain patterns or trends. These patterns can be used by organizations for

making many business-critical decisions. Data analysis has been there for many years; however, due to the recent IT boom, the requirement of having a more efficient analytical system has increased a lot. Data is being generated at a very rapid rate. Data generated by the end user using an eCommerce site can be used to identify different buying patterns of the consumers residing across geographies. Many government agencies analyze data generated by citizens for making citizen-friendly decisions and policies. For analyzing this data, this data needs to be put in some place from where it can be queried or used. The data warehouse is one such service.

Data Warehouse

Data warehouse is a relational database management system that stores data coming from various locations. This data can be used for performing analysis by running SQL-like queries. You can also use this data for performing business intelligence and reporting. Raw data is loaded into a Data Warehouse after performing required cleansing and transformational activities so that it can be brought into SQL-like format.

OLAP versus OLTP

OLTP stands for online transactional processing, whereas OLAP stands for online analytical processing. OLTP systems are used for storing and maintaining transactional processing data using various database services. An OLTP database stores data in multiple fields and records. OLTP databases are typically used for performing insert, update, and delete operations, and they do not require historical data. OLAP systems store data in Data warehouse systems. Data warehouses are primarily designed to facilitate searches and analyses and usually contain large amounts of historical data. Data stored in a Data Warehouse system is used for decision-making, Business intelligence, planning, and analyzing. Some examples of Data Warehouse systems are DataMart, IBM DB2 warehouse, GCP BigQuery, and so on. Examples of OLTP systems are MySQL, MSSQL, Oracle, and so on. A data warehouse takes inputs from multiple resources and stores them in one location that is used for performing various analytical activities.

*Table 7.1*makes a comparison between OLTP and OLAP systems:

OLTP	OLAP
OLTP stands for Online transactional processing.	OLAP stands for Online analytical processing.

OLTP processing is faster, typically in a few milliseconds.	OLAP systems can take from seconds to minutes, depending on the data that needs to be analyzed.
OLTP systems mostly run INSERT, UPDATE, and DELETE queries.	OLAP systems mostly work on SELECT queries.
OLTP systems are usually designed for specific industries such as retail, manufacturing, finance, and so on.	OLAP systems are designed for a specific subject like sales marketing and so on.
It mostly works mostly on live data.	It works mostly on historical data
MySQL, MSSQL, Oracle, and so on are a few examples of OLTP systems.	Oracle DataMart, GCP BigQuery, IBM DB2 warehouse, and so on are a few examples of OLAP systems.
OLTP systems usually have shorter transactions.	OLAP systems have larger transactions as the transactional output is used for analyzing a large amount of data.

Table 7.1: *OLTP versus OLAP*

Now, let us discuss about the popular GCP native Data Warehouse service called BigQuery.

BigQuery

We have already discussed briefly about BigQuery in *Chapter 4, Database services in GCP*. Now, we will try to explore a little more about BigQuery. When it comes to analytics, data warehouse plays a very important part. After the incoming batch and streaming data are processed, they can be ingested into a data warehouse for performing required analytics. BigQuery is a data warehouse service offered by GCP that can be used to perform analytics by running SQL-like queries. This service can easily get integrated with data visualization tools like Looker or PowerBI. Data visualization tools are used for a graphical representation of data after analysis is done.

BigQuery architecture

BigQuery decouples compute from storage. Because of this decoupling, compute nodes and storage can scale independently from each other. The major BigQuery components are as follows:

- **Dremel**: Dremel is an execution engine in BigQuery. Dremel consists of "*leaves*" and "*Mixer*". Together they are part of the execution tree. Any incoming query is converted into an execution tree. "*Leaves*" are

responsible for reading data from storage and doing all computational work required for running queries. "*Mixer*" does require aggregation. Dremel assigns required slots for executing these queries. Depending upon the query, Dremel can assign multiple slots.

- **Colossus**: Colossus is storage in BigQuery. It is a proprietary file system from Google. It is a next-generation distributed file system used by many GCP services such as Cloud Spanner, Cloud Storage, and so on. Colossus clusters provide a dedicated disk to the users to run the BigQuery. It is scalable storage, so it can span upto several petabytes. It also handles replication between clusters.
- **Jupiter network**: Jupiter network enables the communication between compute and storage.
- **Borg**: Borg is responsible for allocating hardware resources for executing SQL queries. Borg is a cluster management system that has many machines with multiple cores. Whenever any query needs to get executed, Borg assigns a fraction of compute or processing power of a cluster to execute those queries.

Figure 7.9 shows the various architectural components that form BigQuery:

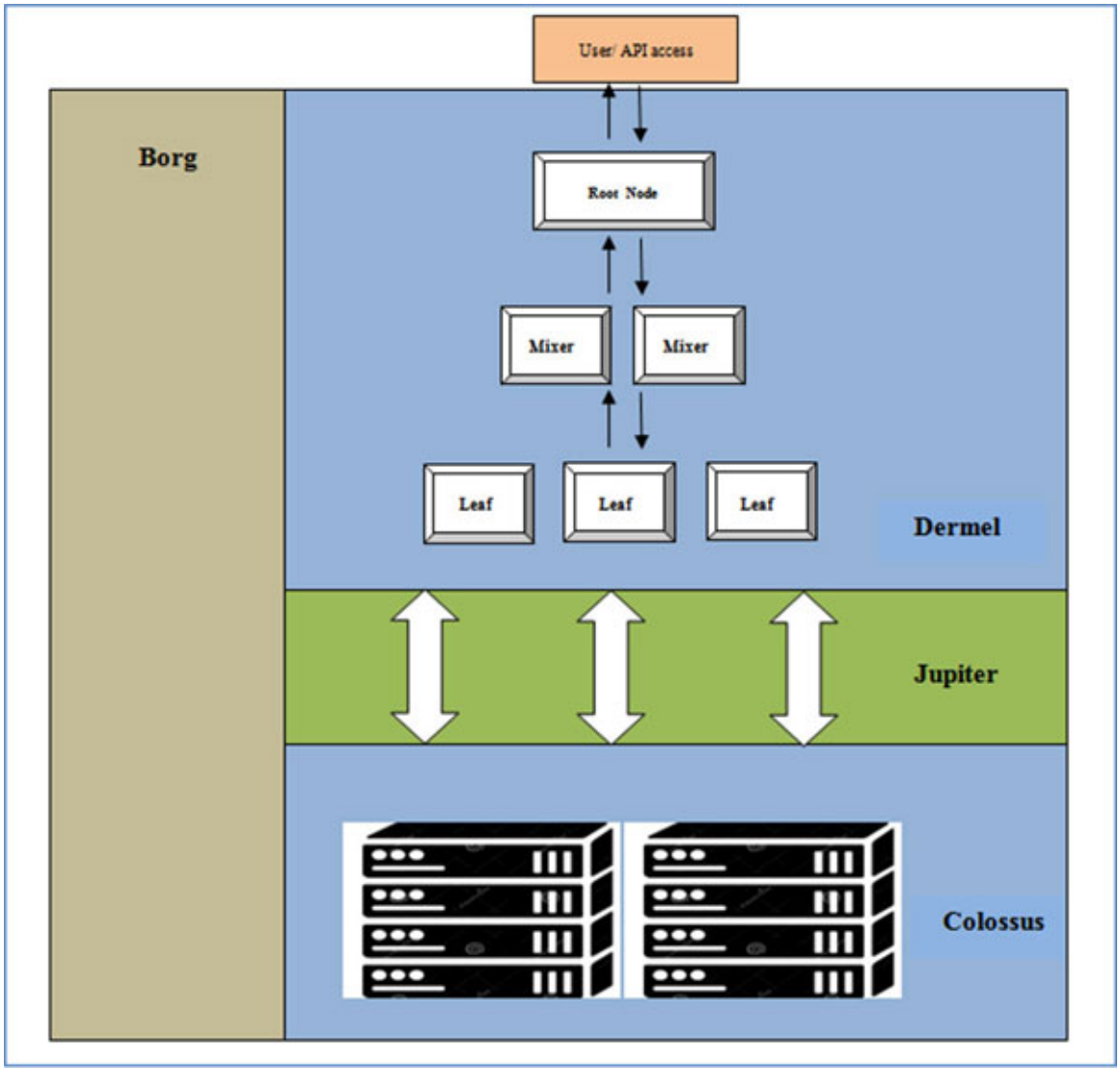

Figure 7.9: *BigQuery architecture*

Storage management in BigQuery

BigQuery organizes the data in projects, datasets, and tables. Each dataset belongs to a project. Each project can have multiple datasets, and each dataset can contain multiple tables. Datasets can be regional or multi-regional. Tables are a collection of rows and columns that are defined by the schema. When you do not want to perform operations on actual tables, you can make use of views. Views are the virtual tables that get created by using SQL queries and provide access at the view level. You can run Jobs for performing any specific activity. You can run jobs for copying data, loading data, exporting data, or querying data. You use standard SQL queries for Querying stored data in

BigQuery. As SQL is not part of the scope, we will not be covering that in this book. Readers are advised to understand the SQL concepts to develop more understanding of them.

Any traditional relational database management system is row-oriented, which makes it possible to provide fast transactional processing, whereas data warehouse systems like BigQuery stores the data in columnar storage, which is good for analytical processing. Every column in BigQuery is stored as a separate file block. BigQuery uses a proprietary columnar format to store the column data in files. This proprietary columnar format is called a **capacitor**. Refer to *figure 7.10*:

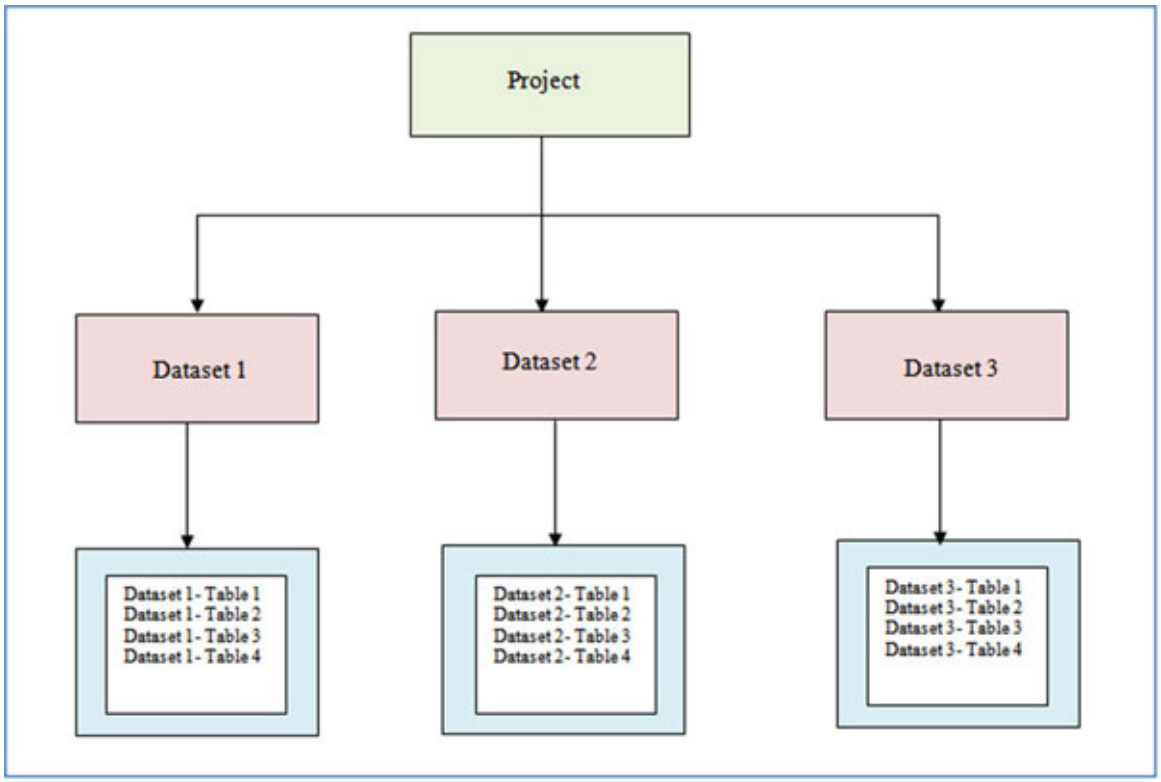

Figure 7.10: *Storage management in BigQuery*

Partitioning and clustering in BigQuery

Partitioning is used to divide a large table into smaller partitions. You can partition a table using a timestamp or any other integer value. Partitioning makes it possible to segment large tables into smaller segments which results in improved latency. BigQuery uses columnar-based partitioning. If you have enabled columnar portioning in a table, the data written to the table will be automatically portioning to the table.

Figure 7.11 an example of columnar portioning:

Date	Product	Sale	Customer
2022-03-01	Gin	2	Vikas
2022-03-01	Whiskey	3	Srivathsan
2022-03-02	Beer	3	Dheeraj
2022-03-02	Whiskey	3	Mohit
2022-03-02	Beer	2	Sushan
2022-03-03	Rum	3	Sunil
2022-03-03	Vodka	3	Parth
2022-03-03	Beer	4	Praveen

Columnar partitioning

2022-03-01

Date	Product	Sale	Customer
2022-03-01	Gin	2	Vikas
2022-03-01	Whiskey	3	Srivathsan

2022-03-02

Date	Product	Sale	Customer
2022-03-02	Gin	2	Vikas
2022-03-02	Whiskey	3	Srivathsan
2022-03-02	Beer	2	Sushan

2022-03-03

Date	Product	Sale	Customer
2022-03-03	Rum	3	Sunil
2022-03-03	Vodka	3	Parth Sharma
2022-03-03	Beer	4	Praveen

Figure 7.11: *Columnar partitioning*

Now let us discuss about clustering in BigQuery. Clustering helps with organizing the data as per the content of one or more columns in a table. Once clustering is enabled in a table, the newly added data automatically gets sorted according to the value in the clustered column or columns. *Figure 7.12* shows an example of clustering:

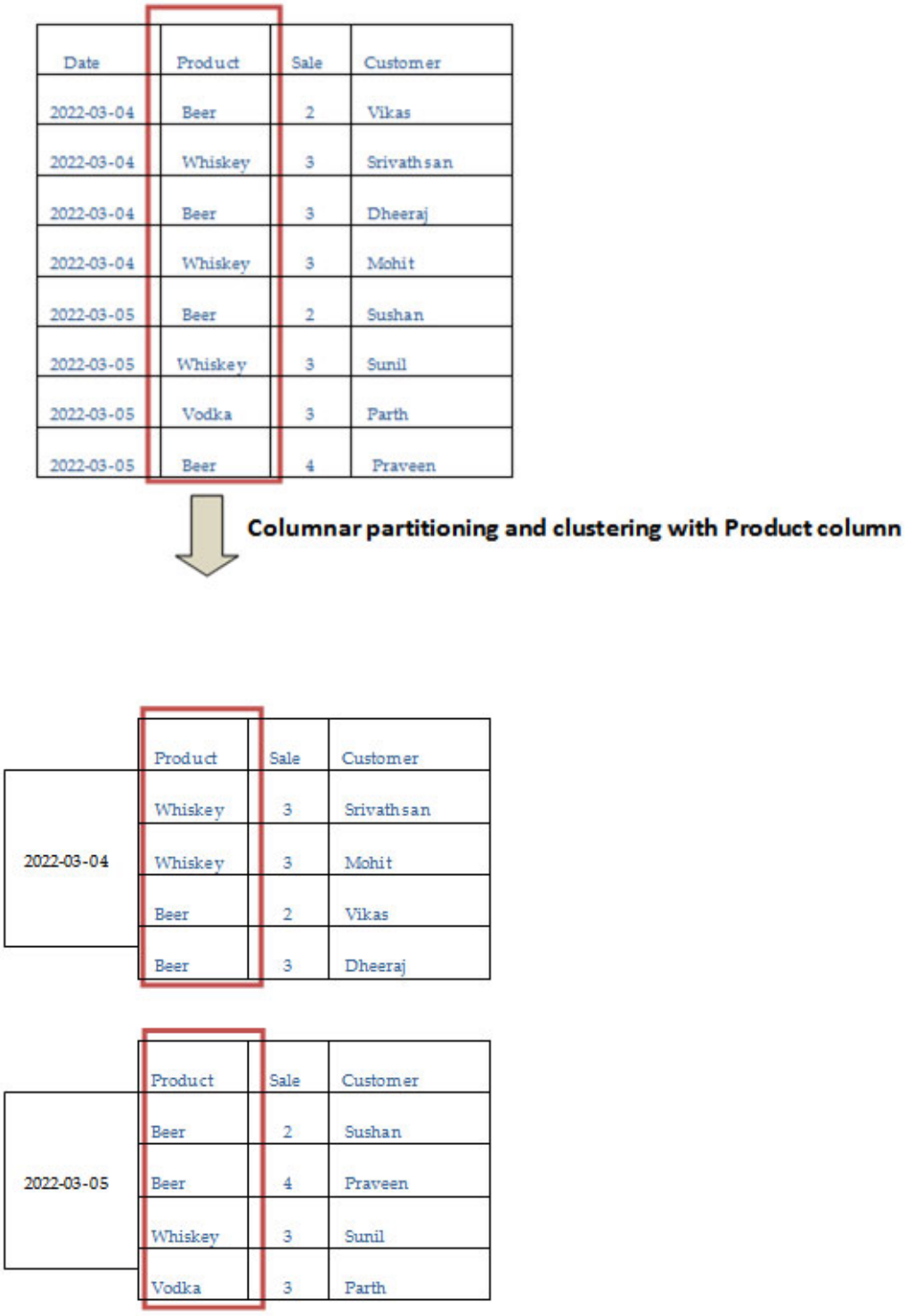

Figure 7.12: *Columnar partitioning with clustering*

Clustering helps in cases when you use a filter against a specific column while running a query, as it does not require you to scan the entire. It also helps while

running some aggregation against values in a specific column.

Data visualization tools in GCP

Data visualization tools in GCP collect the data, analyze it and represent it in the form of charts, maps, diagram, and so on that can be utilized for making critical business decisions. These tools can get integrated with data warehouse solutions to show the desired output. Some examples of data visualization tools are PowerBI, Tableau, and so on. Google cloud Looker studio is the main data visualization and business intelligence tool offered by GCP.

Looker Studio

Looker Studio is an offering by Google Cloud Platform that connects with various data sources and helps in creating interactive dashboards, charts, and reports. These dashboards can be used for making important business decisions. It can easily get integrated with services such as Big Query, Cloud Storage, App Sheets, and so on. Looker Studio offers many connectors created by Google, Partners, and the community for integration with various data sources. It also gives you the option to build your own connectors from scratch, using which you can connect with various other data sources. It provides support for SQL and non-SQL data sources. When data is coming from multiple sources, you can merge that data and create visualization dashboards using Looker Studio. It is simple to use and provides various customization options. Looker Studio comes in two varieties- Looker Studio and Looker Studio Pro. Looker Studio is a free version, whereas Looker Studio Pro includes all features of Looker Studio plus enhanced enterprise capabilities and technical support. Using Looker Studio Pro, it is easy to manage access to data sources and reports with features like team workspace and Google cloud project linking.

A typical example of Looker studio involves the following steps:

1. Create a Query using BigQuery on a dataset.
2. View table schema in BigQuery.
3. Visualize the results in the Looker Studio report.

Following are some of the important features of Looker Studio:

- **Granular insight into data**: Looker Studio is powerful and can help in filtering the data in the dashboard to look at it at a very granular level.
- **Browser-based**: Looker Studio is browser-based; therefore, no installation is required.
- **Looker Studio API**: Looker API can be used while building custom applications and workflows. It provides additional APIs to interact with the dashboard.
- **Report Embedding**: In Looker studio, you can include your Looker Studio report on any Web page. Hence, you can share your data story with anyone through your Web page.
- **User-friendly interface**: Looker Studio provides easy to use interface. It has a report editor feature that provides you drag-and-drop functionality.

Google Cloud Cortex Framework

Google Cloud Cortex framework is the latest offering from GCP, which includes various reference architectures, accelerator services, and deployment templates that help in connecting and launching analytics solutions from SAP and Non-SAP sources and help in reducing the time and effort required to build solutions. Cortex Data Foundation for SAP helps with faster insights of SAP data into BigQuery. It provides a reference architecture, already defined views, and CDC scripts that help build solutions in a fast and efficient way.

Cortex data integration service helps with integrating SAP and Non-SAP data into BigQuery. Some example of Non-SAP data sources is Google Trends, Weather, and so on.

The cortex application layer also helps with exposing Cortex foundation results to external applications through production-ready APIs and Cloud Pub/Sub services.

Google Cloud Cortex framework is a relatively new offering and is expected to get better with time. Readers are advised to go through Google's official documentation link mentioned as follows to develop more understanding of it: **https://cloud.google.com/solutions/cortex**

Reference architecture showing Streaming and Batch data coming from different sources and

utilizing various data services in GCP

Refer to *figure 7.13*, which shows Batch data coming from different sources and using various data services in GCP:

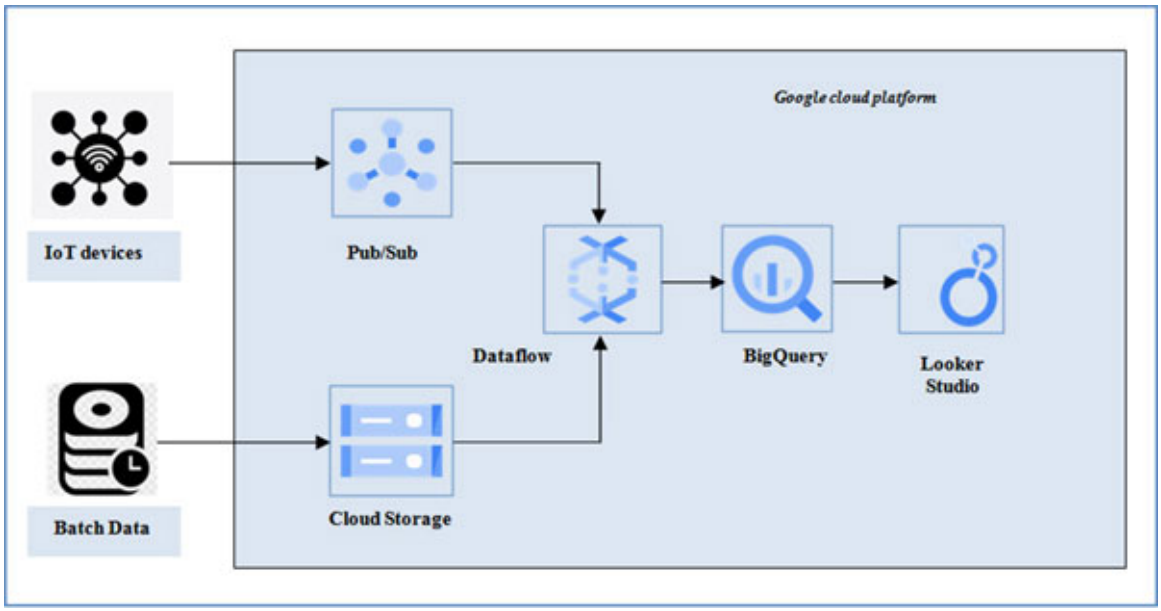

Figure 7.13: *Batch and stream processing example*

The preceding example shows a high-level architecture of the stream and batch data transformation pipeline. For stream processing, we have shown IoT devices generating streaming data, and for batch processing, we have some external storage devices where historical data is stored.

The preceding pipeline can process the data in the following sequence:

1. For IoT devices, data ingestion happens through Cloud Pub/Sub. As discussed earlier, it is a messaging service that takes the streaming data as input and delivers it to a subscriber service through topics. The subscriber subscribes to a topic through which this message is received. In this example, Cloud Dataflow works as a subscriber.
2. Batch data coming from external storage is ingested into Google Cloud Storage through gsutil. You can install gsutil client into source storage and run commands or batch jobs to copy historical data into the Google Cloud Storage service. This Cloud Storage is integrated with Dataflow.
3. Dataflow performs the required transformation on this incoming batch and stream data.

4. BigQuery service is integrated with Dataflow and takes this data from Dataflow in the form of Datasets.
5. BigQuery is integrated with Business intelligence and visualization tool like Looker. Various queries are run on BigQuery tables to produce the desired output on which analytics can be performed.
6. Looker Studio helps with creating various visualization dashboards to produce the desired results. This output is then used for making business critical decisions.

Conclusion

In this chapter, we have discussed about various data services available in the Google Cloud Platform. Data transformation, analytics, and visualization, are in-demand skills. Data engineers and data scientists who have some working knowledge of the Google Cloud Platform are always in demand. Readers are advised to go through the GCP site and other online study material to develop a further understanding of this. In the upcoming chapter, AI/ML in Google cloud, we will understand basic concepts of AI and machine learning and will also discuss about various AI/ML services available in the Google Cloud Platform.

Key terms

- **CDAP**: Cask Data Application Platform—open-source application development platform for the Hadoop ecosystem.
- **IoT**: Internet of things.
- **SAP**: systems, applications, and products. Software products developed by the company with the same name help organization logistics, financials, human resources, and other needs.
- **Machine Learning**: Machine learning is a part of Data Science and AI that focuses on developing algorithms that try to imitate the self-learning capability of the human brain.

Questions

1. What do you understand by Big Data? State the difference between structured and unstructured data.

2. What is the difference between stream and batch processing?
3. What is a messaging service? Explain Pub/Sub service in GCP.
4. What is the difference between Dataflow and Dataproc services in GCP?
5. What is a Data warehouse service, and why it is used? Explain various features of the GCP BigQuery service.
6. What are data visualization tools? Describe Cloud Looker Studio.

Further readings

- **https://cloud.google.com/Dataflow**
- **https://cloud.google.com/Dataproc**
- **https://cloud.google.com/pubsub**
- **https://cloud.google.com/BigQuery**
- **https://cloud.google.com/looker-studio**

CHAPTER 8

AI/ML in Google Cloud

Machine Learning(ML) is a subfield of Artificial Intelligence (AI). The goal of ML is to make computers learn from the data that you give them. Instead of writing code that describes the action the computer should take, your code provides an algorithm that adapts based on examples of intended behavior. The resulting program, consisting of the algorithm and associated learned parameters, is called a trained model.

-Google

Introduction

Human intelligence has the capability to learn from the external environment and evolve accordingly. In Artificial Intelligence, we try to simulate the same human intelligence and apply it to the machines. In Machine Learning, we try to develop computer systems that can learn from the data that is fed to them and perform given tasks more efficiently. As part of this, we design and create algorithms that take data as input and learn from it. This results in more efficient and better-performing systems.

Structure

In this chapter, we will cover the following topics:

- What is Machine Learning?
- Machine learning in Google cloud
    - Custom models in ML
    - Pre-trained models in ML
- Various AI tools and services in Google cloud
    - Vertex AI
    - AutoML

    - Natural Language API
    - Speech-to-Text API
    - Text-to-Speech API
    - Dialogflow AI
    - Translation AI
    - Cloud vision API
- Neural network
- TensorFlow
- Deploying and training ML model in GCP

Objectives

After reading this chapter, you will be able to understand the basics of Artificial Intelligence and Machine Learning and the services that can be leveraged in the Google Cloud Platform for building Machine Learning models. AI/ML are niche skills that are in high demand and require coding and scripting experience. Readers are advised to enroll in a specialization program if they want to consider AI/ML as a career option.

What is Machine Learning?

Machine Learning is a subset or branch of Artificial Intelligence that focuses on building systems that learn from the data that is fed into them. At present, we can see several examples of Machine Learning around us. Some examples include Machine Learning algorithms that are being used to make your online shopping experience smooth and efficient or the algorithms used by social media websites that make it possible to show you the content as per your interests. These models take input data through human or non-human interactions and get trained or can fine-tune themselves to produce the desired results that are more accurate. Refer to *figure 8.1*:

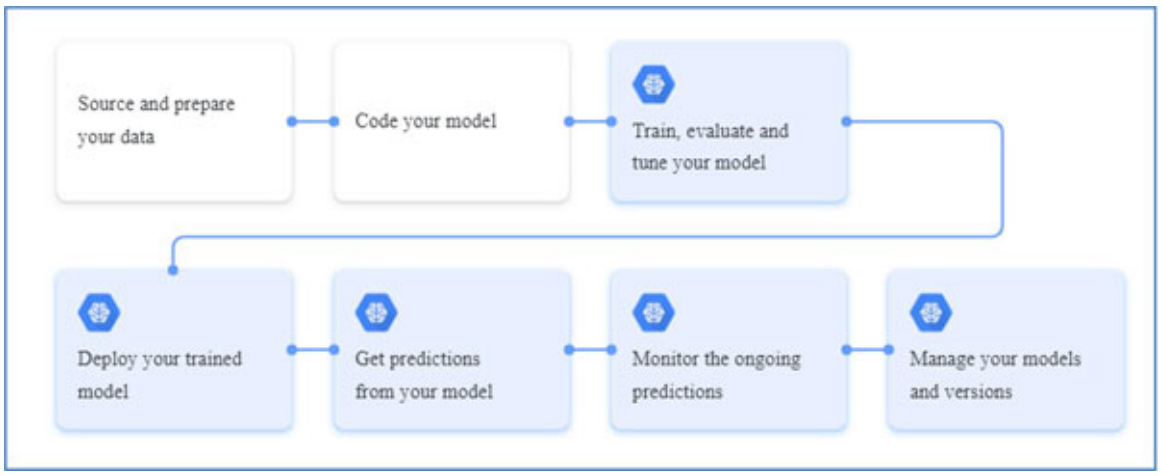

Figure 8.1: *Machine Learning workflow*

Figure 8.1 depicts the high-level overview of the Machine Learning workflow. When the data is sourced and prepared, any requirements specific to cleansing and transforming the data are fulfilled. This data is used to train the Machine Learning models. As it is seen clearly, Machine Learning requires access to a large amount of training data to train the model. More training data results in better accuracy of the model.

Machine Learning approaches

There are two basic techniques for ML. They are Supervised and Unsupervised Machine Learning. Apart from the preceding two techniques, there is a third type called Reinforcement Learning. All three techniques are explained as follows:

Supervised Machine Learning

As the name suggests, supervised Machine Learnings a method in which Machine Learning algorithms learn by using training datasets that are labeled and tagged with the right output. In this type of learning, iterative predictions are made, and after checking the output, required adjustments are made to make the correct predictions. In this method, we try to find a function that can map the input data to the output data.

$$Y=f(X)$$

**Y is the output data, X is the input dataderived from the dataset, and f() is a function.*

Supervised learning can help in solving two types of problems:

- **Classification**: The classification problem is the one in which we need to categorize or classify the output result. For example, if you have many fruits in a basket and you need to classify them into red and yellow, or you have some species in a picture, and you need to categorize them into birds and animals.
- **Regression**: Regression focuses on values rather than categorization or classification. Some of the examples include predicting the price of some property in some area and predicting the price of some fruit at a specific time of the year. There are mainly two types of regression, i.e., Linear Regression and Logistic Regression. Linear Regression is used to predict the value of an output variable depending on the input variable, whereas Logistic Regression defines the category of an output variable depending on the value of the input variable.

Supervised Machine Learning examples

Following are a few examples of supervised learning:

- Text recognition
- Sentiment analysis
- Decision trees
- Image recognition
- Stock predictions
- Weather predictions

Unsupervised Machine Learning

Contrary to supervised learning, unsupervised learning takes a more independent approach. In unsupervised learning, input data is not labeled. Multiple predictions are made in order to come up with the most accurate result. Unlike supervised learning, unsupervised learning requires minimum human intervention to predict the output. The only time unsupervised Machine Learning requires human intervention is during the time when output needs to be verified. Unsupervised learning requires huge amount of data to train the models and provide the most accurate output.

Unsupervised learning can be understood by a child learning analogy. When a child identifies fruits by observing their shape and color instead of memorizing them, it is analogous to unsupervised learning.

Unsupervised Machine Learning methods

Unsupervised Machine Learning can use the following methods to solve a problem:

- **Clustering**: In clustering, the Machine Learning model tries to identify similarities and differences between data objects and categorizes them accordingly. Objects with the most similarities remain part of a group.
- **Association**: Association is another unsupervised Machine Learning method that is used to identify the relationship between different variables in a large dataset and map them accordingly. One of the use cases of the association method is Web-based mining, where you need to find an association between various elements like website URLs, hyperlinks, and access logs. Another example is to identify buying pattern of a customer by associating item type, retailer names, and so on. A variable is a state or value that can be modified by performing operations on it.

Reinforcement Machine Learning

Reinforcement Machine Learnings the third type of Machine Learning method. This Machine learning is based on the reward and punishment concept. In this Machine Learning technique, an intelligent agent learns based on the feedback it receives on performing a specific action. For any right action, the agent receives positive feedback, whereas for any negative action agent receives negative feedback. The agent learns from this feedback and evolves accordingly. There is no labeled data used in reinforcement Machine Learning, and its goal is to enhance performance by learning from positive feedback. For any negative feedback, the agent gets penalized. For example, in a board game, the Reinforcement Learning algorithm can learn from the results that are produced by making different moves. This learning method works in a delayed return. There are no predefined labels, and hence, there are possibilities for exploring different ways of completing any task.

Machine Learning in Google Cloud

Google Cloud provides a number of AI/ML services that can be leveraged. In GCP, you can either build your ML models from scratch, or you can use some pre-existing pre-trained models as described as follows:

Custom models in GCP

GCP provides tools and services that can be used to build and train AI/ML models. These models are created for specific requirements and trained by feeding input data to generate a more optimized or more accurate output. Developing a custom model requires you to source and prepare the training data. This data is then fed to the Machine Learning model for training. It is suggested to feed a large amount of training data to an ML model to maximize the chances of accuracy. Vertex AI and Cloud AutoML are some of the services offered by GCP that can be used for building and training custom models. We will discuss about various AI/ML tools and services on the topic of various AI options in Google cloud.

Pre-trained models in GCP

GCP provides many ready-to-use ML models that can be reused for various requirements. Google provides many services exposed through APIs around these pre-trained ML models that can be integrated with your custom-built application without the need to develop or train the ML models. As these ML models have evolved over a period of time and have been a part of the Google ecosystem for a long time, they are very much reliable. Google Cloud Translation API, Cloud Vision API, Cloud Speech-To-Text API, and so on are some APIs that use pre-trained machine learning models to produce the desired results.

Various AI tools and services in Google Cloud

Following are some of the AI products and services offered by the Google Cloud Platform.

Vertex AI

Vertex AI is a unified Machine Learning platform offered by GCP that can be used to build, deploy and scale AI models effectively. With Vertex AI, you can build your AI models from scratch or use one of the pre-trained APIs available. Vertex AI is integrated with various tools and services offered by GCP that help you with creating, training, and deploying ML models. For example, you can create ML models using BigQuery ML, or you can export BigQuery datasets to Vertex AI workbench to run your models. Vertex AI also provides

support for many open-source platforms, such as TensorFlow and PyTorch, by providing seamless integration. Vertex AI brings AutoML and AI Platform together into a unified API, client library, and user interface.

AutoML

AutoML is another offering from the Google Cloud Platform that helps developers to build and train Machine Learning models with limited expertise. AutoML has the following line of products:

- **AutoML Natural Language**: AutoML natural language has offered, such as AutoML text and document classification, entity extraction, and sentiment analysis.
- **AutoML tables**: AutoML tables allow you to build Machine Learning models on structured data with speed and scalability.
- **AutoML translation**: AutoML translation helps you with creating customer translation models.
- **AutoML Video intelligence**: It helps with training your Machine Learning models to classify shots in your videos and is also used for object tracking in the videos from one shot to another.
- **AutoML vision**: AutoML vision is used to help you to train your customer machine models by helping you with image classification and object detection within an image.

AutoML allows you to train models on datasets without the need to write any code, whereas using an AI platform, you can run your custom training code.

Natural language API

Natural language API is another offering from GCP that provides you with powerful pre-trained Machine Learning models. This AI can be used by developers for sentiment analysis, syntax analysis, content classification, entity analysis, and so on. This AI helps you with classifying a document into 700+ different categories while doing a content classification. It provides an API that can easily get integrated with your applications. It can also identify entities within a document and categorize them under various categories. It can analyze text in multiple languages such as English, Japanese, Chinese, French, German, and so on.

Speech-To-Text API

Google's Speech-To-TextAPI is widely used for converting speech into text with accuracy. It provides many important features like support for global vocabulary, streaming speech recognition, Speech-to-Text On-Prem, content filtering, and so on. Speech-To-Text AI provides support for domain-specific pre-trained models, for example, video, phone calls, medical conversations, long and short utterances, and so on. It also allows you to pass any audio file as URI and convert it into text output. Some examples include textbooks, contact center AI, adding subtitles to your multimedia content, and so on. Refer to *figure 8.2*:

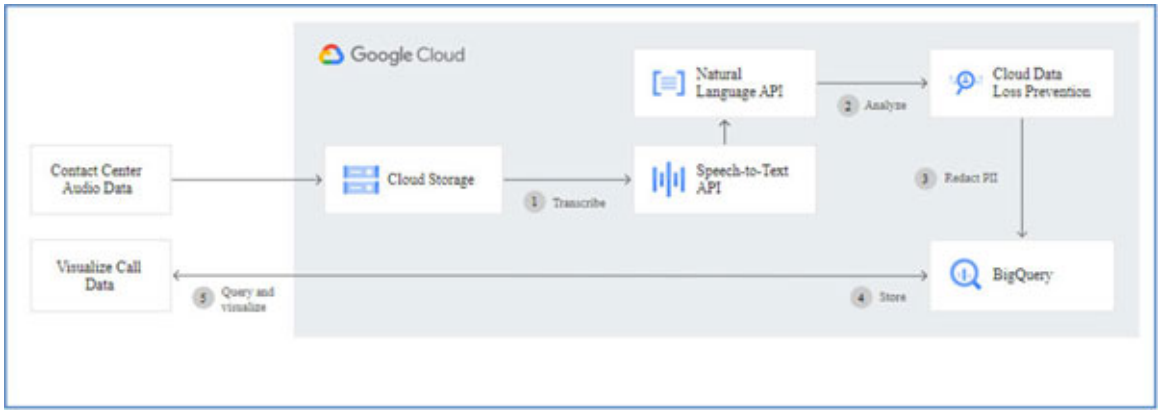

Figure 8.2: *Speech-To-Text API example*
Source: *https://cloud.google.com/speech-to-text*

Text-To-Speech API

Text-To-Speech API is used to convert text into natural-sounding speeches. It provides more than 100 voices to choose from. It uses WaveNet neural network to generate human-like speeches. It also allows you to adjust the pitch of the selected voice and provide you an option to apply **Speech Synthesis Markup Language** (**SSML**) tags, using which you can customize speech for pause and formatting of date, time abbreviations, and so on. As of date, Text-To-Speech API supports more than 40 languages. In the future, you will also get an option to choose from a custom voice model that can be unique for your organization. Please note that the custom voice option is in the beta testing phase right now. Some examples of Text-To-Speech API are Voice bots, voice generation, and so on.

Dialogflow

Dialogflow is a natural language conversational platform that can integrate with Web or mobile applications. Dialogflow takes input from the users in the form of voice or text messages and responds to them in the form of speech or text. Dialogflow CX and ES are two virtual agent services. These services have their own agent types, API, client libraries, and documentation. Dialogflow CX is an advanced agent type that is suitable for large and complex requirements, whereas Dialogflow ES is a standard agent that is suitable for small and simple requirements.

Translation AI

Translation AI is also one of the important AI from Google. This AI can detect and translate more than a hundred languages as per your requirement. Google provides the following products that can be used as per your requirements:

- **AutoML Translation**: AutoML Translation helps in training custom models developed by experts. You can integrate AutoML translation with your custom model and upload translation pairs. AutoML Translation model will help in training your custom models as per your domain-specific needs.
- **Translation API**: Translation API can translate any text in more than 100 languages. You can integrate this API with your custom build applications. Translation API can be customized as per your domain-specific needs, which can help in translating specific terms and phrases.
- **Media Translation API**: Media translation API is another important API provided by Google. This API can do a real-time audio translation. You can integrate this API with your application to convert your Audio content with enhanced accuracy with low latency.
- **Translation Hub**: Documentation translation service provided by Google that is used to translate a large volume of the document into many different languages.

Cloud Vision API

This API uses **optical text recognition** (**OCR**) to detect text from the images and then annotates an image based on the text that is detected. It is also for face and landmark detection, image labeling, and tagging for explicit content.

Neural Network

Neural Network or artificial neural network is a system that consists of multiple algorithms functioning together, mimicking the human brain. The neural network is part of Machine Learning and helps in creating Machine Learning models. Refer to *figure 8.3*:

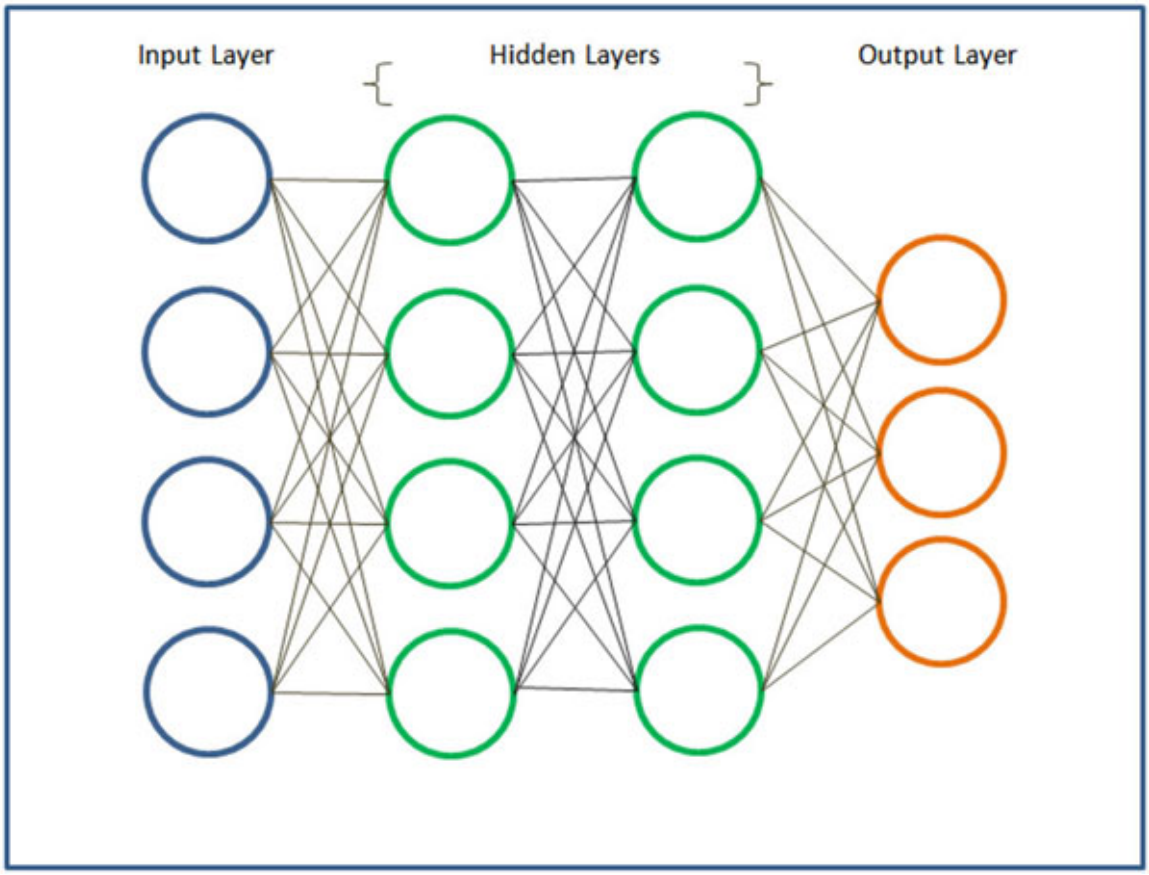

Figure 8.3: *Neural Network*

The following are the various components of a neural network:

- **Neurons**: A Neuron, or more precisely an artificial neuron is a node that receives an input signal, processes it, and sends it to other neurons that are connected to it. This input has a number or value associated with it. A neuron computes these inputs using some non-linear function and provides the required output.
- **Connections or Edges**: Connections or edges are something through which different nodes or neurons connect with each other and transmit signals. Neurons and edges have a weight. This weight is readjusted with

the learning. It can increase or decrease, and accordingly strength of the signal also increases and decreases.

- **Layers:** The aggregation of the neurons is called layer. The following are three main types of layers:
    - **Input Layer**: Input Layer is the one through which all inputs are fed to the neural network.
    - **Hidden Layer**: Hidden layer is the one that resides between the input and output layer. This layer processes the signal/data and sends it to the output layer. A neural network can have multiple hidden layers. It takes the input signal with weights and processes it to produce the output.
    - **Output Layer**: Output layer is the final layer that represents the output of the neural network. It takes input from the hidden layer and presents it as an output of the neural network.

Artificial neurons connect with other neurons and have some threshold and value associated with them. If the output generated by one specific node is above that threshold, the value is passed to the next layer. As it can be understood, the quality of the output generated by a neural network depends on how much training data is fed to it.

Types of neural networks

Following are the major Neural network types:

- **Perceptron**: Perceptron is a simple neural network that accepts weighted inputs, does computation, or applies some activation function on it to generate output. Perceptron is a supervised learning technique that classifies the data into two categories. Therefore, itis mainly used for implementing And, OR, and NAND logics and used for binary operations. The activation function is a non-linear transformation done over an input before sending it to the next layer. Refer to *figure 8.4*:

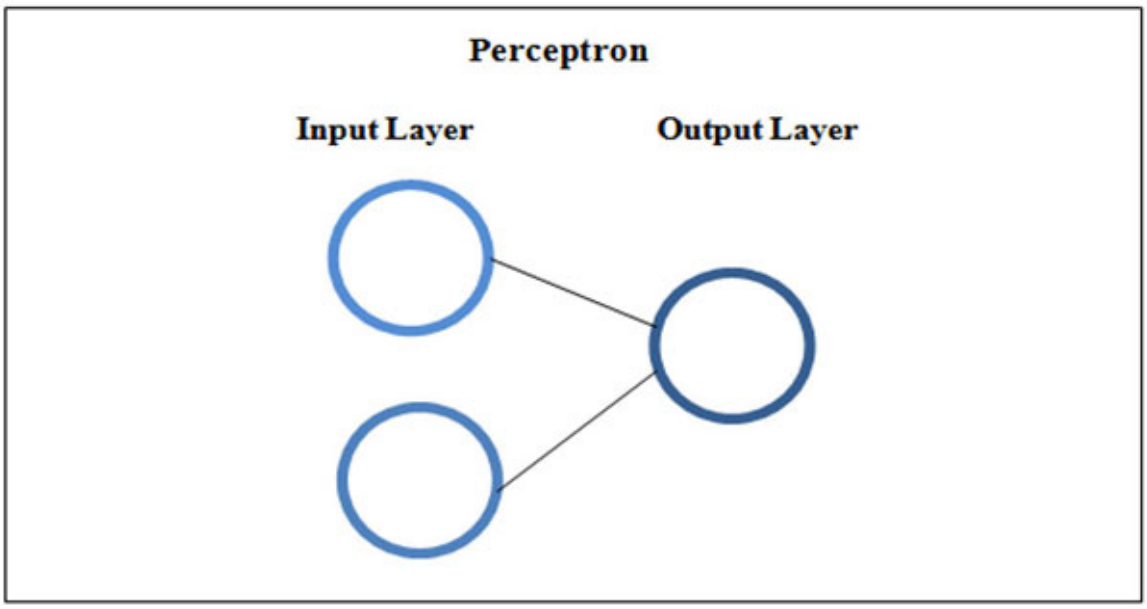

Figure 8.4: *Perceptron*

- **Feed Forward Neural Networks**: Feed forward is one of the simplest types of neural networks in which the data travels in only one direction. This type of neural network can be single-layered or multi-layered. Since this neural network is uni-directional, weight is a static entity in this neural network. This neural network can be used in areas such as face recognition, speech recognition, and so on. Refer to *figure 8.5*:

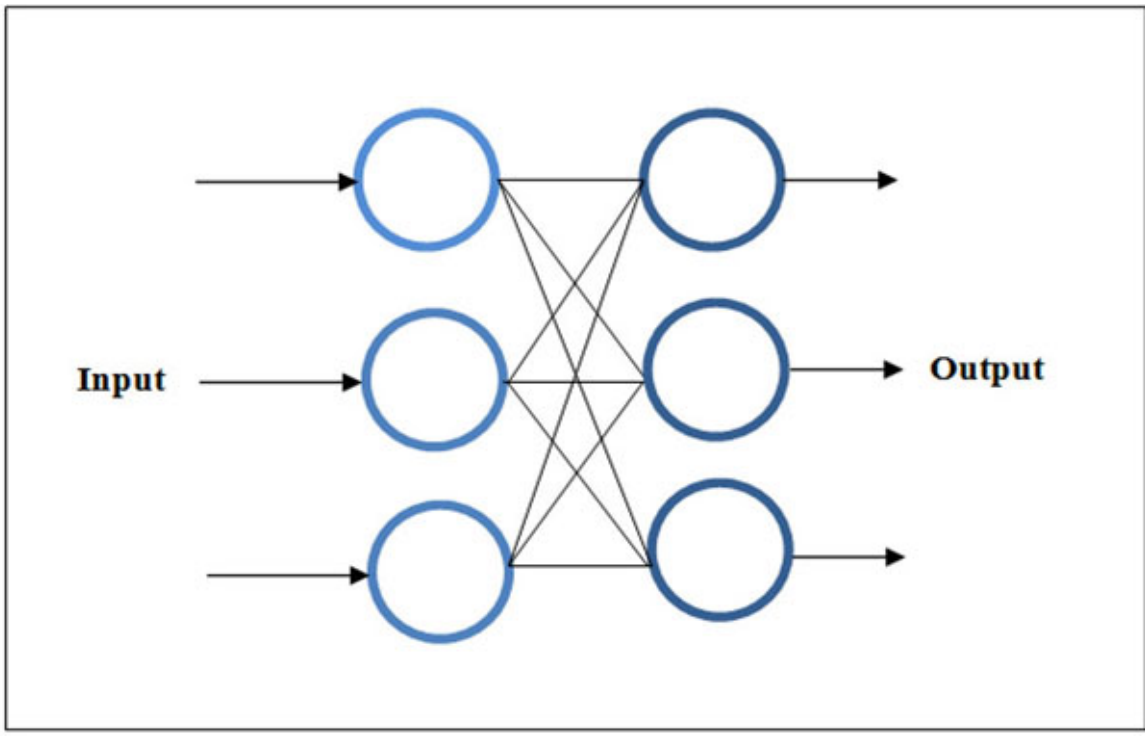

Figure 8.5: *Feed forward*

- **Multilayer perceptron**: Multilayer perceptron is a complex neural network. It includes an input layer, one or more hidden layers, and an output layer. In this type of neural network, one neuron is connected to all neurons in the next layer. Weights and inputs are multiplied together and fed to the activation function. The neurons in multilayer perceptron are trained with the back propagation learning algorithm. This algorithm back propagates errors from output nodes back to the input nodes. It supports bi-directional propagation. The multilayer perceptron is used in deep learning and is complex to develop. They are used for solving complex classification problems.
- **Recurrent Neural Network**: Recurrent neural network is another interesting neural network type. The first layer is a feed-forward neural network and then has recurrent neural networks. RNN uses a concept of a feedback loop where input in the previous step is retained in the memory. In RNN, a feedback loop connection allows hidden layers at a one-time instance to be used as input to the next time instance. Recurrent Neural networks can be used for Text-To-Speech processing, sentiment analysis, translation, and so on. Refer to *figure 8.6*:

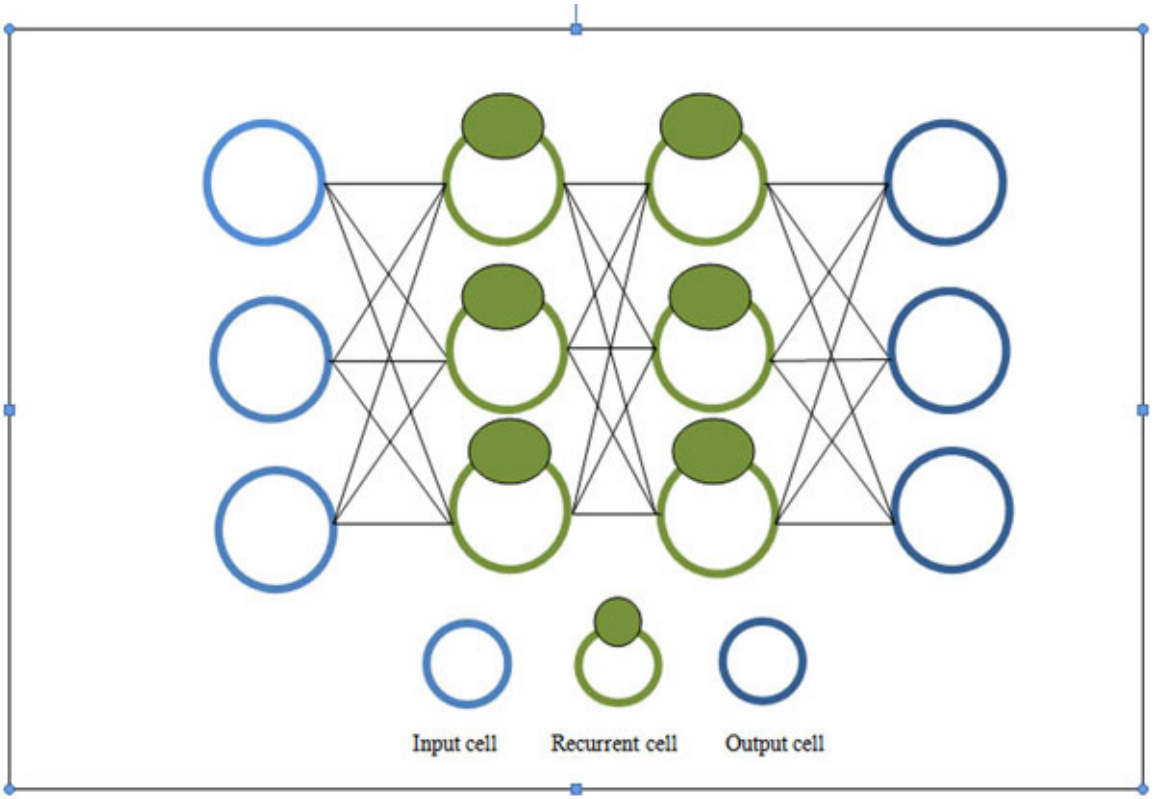

Figure 8.6: *Recurrent neural network*

Convolution Neural Network: Convolution neural network is another popular neural network that is used primarily in image recognition. This neural network requiresthe processing of pixels that helps in identifying the pattern of an image. In CNN, we try to simulate the connectivity pattern of the neurons in the front lobe of the human brain, which is responsible for creating visual stimuli. This neural network identifies weights and biases in the image, which helps in differentiating one image from another image.

Sequence-to-sequence models, Modular Neural Networks, are some other neural network types. Readers are advised to go through some online documentation and videos to develop a further understanding of it.

TensorFlow

TensorFlow is an end-to-end open-source platform that provides various tools, libraries, and other resources for building Machine Learning models. Using TensorFlow, developers can build and deploy Machine Learning models with ease. TensorFlow provides support for many programming languages, such as Python, Java, C++, and so on. It contains workflows and APIs that can be used for building new Machine Learning algorithms and can also train the Machine Learning models. You can build models On-Prem, on Cloud, or on any mobile device. TensorFlow supports many CPU and GPU types. Google also provides (**Tensor Processing Unit**(**TPU**), which is specifically built for Machine Learning and tailored for TensorFlow.

Following is the TensorFlow toolkit Architectural stack. The first layer in the stack is the hardware platform required for TensorFlow. While low-level APIs are used for exploring and building new algorithms, high-level APIs are used for defining and training ML models. We also have a set of reusable libraries that will be used for building ML models. Refer to *figure 8.7*:

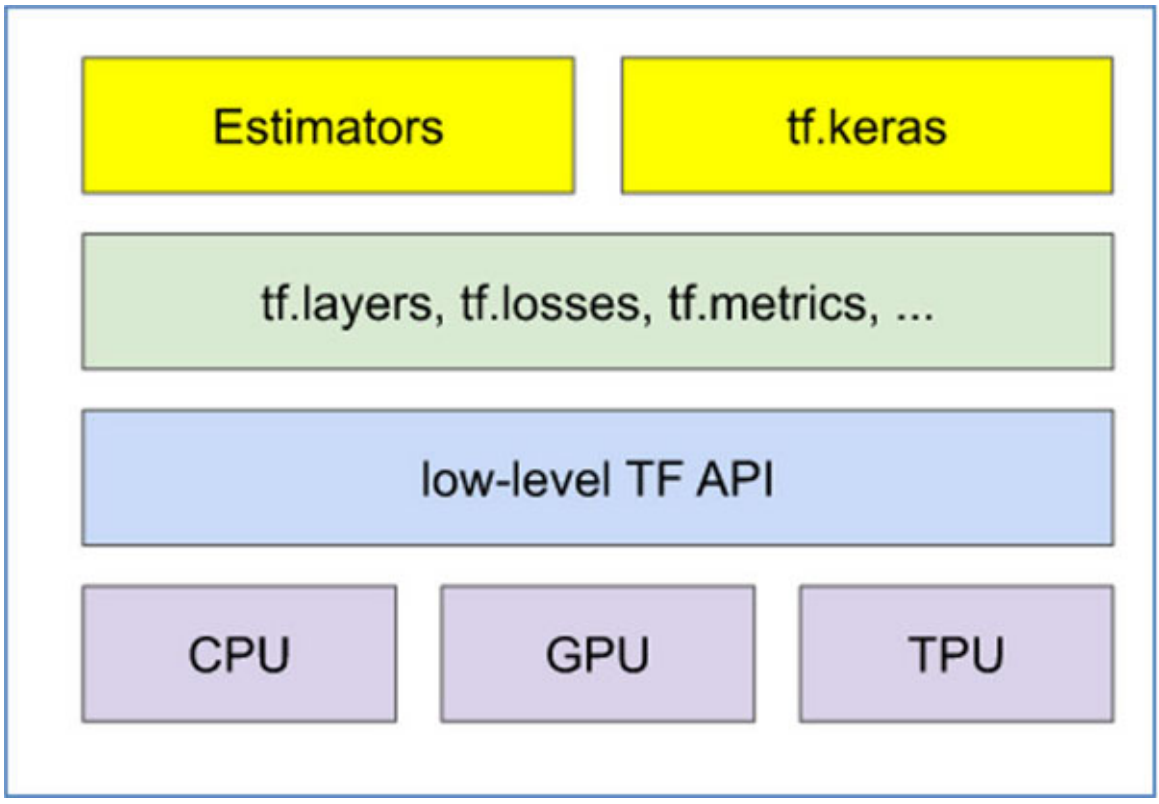

Figure 8.7: *TensorFlow Toolkit.*
Source*: https://developers.google.com/machine-learning/crash-course/first-steps-with-tensorflow/toolkit*

TensorFlow provides an API that supports distributed training across TPUs, GPUs, and multiple machines called nodes. This API is called tf.distribute. Strategy. Following are some of the various strategies by TensorFlow that come under distributed strategy:

- **Mirrored Strategy**: This strategy is used for implementing training on multiple GPUs inside a single machine. In this, training is distributed synchronously across multiple GPUs.
- **Multi Worker Mirrored Strategy**: As the name suggests, this strategy implements training across multiple GPUs inside Multiple worker nodes.
- **Parameter Server Strategy**: The parameter server strategy uses an additional parameter server along with the worker nodes. In this, variables are created in the parameter server that is updated and read by worker nodes.
- **Central Storage Strategy**: In this strategy, variables are not mirrored and placed on a single CPU; however, operations are replicated across multiple GPUs.

- **TPUStrategy**: This is a new strategy that allows users to easily switch their model to using TPUs.
- **One Device Strategy**: Input distributed through this strategy will be prefetched to the specified device.
- **Default Strategy**: Default strategy does not use a distributed strategy.

Deploying and training ML model in GCP

One of the biggest challenges in building Machine Learning models is to set up the environment, which requires libraries, packages, and hardware in place. Google cloud platform provides a service called Vertex AI Workbench that can be used for building these models. This is a fully managed service by GCP that provides a Jupyter Lab environment based on Jupyter Notebook. Jupyter Lab provides the required runtime environment, libraries, and packages that are required to deploy ML models.

We will show how to deploy the ML model through an exercise. In this lab, we will make use of TensorFlow 2.X model training:

1. Go to Navigation Menu and click on **Vertex AI** and then on **Workbench**. Refer to *figure 8.8*:

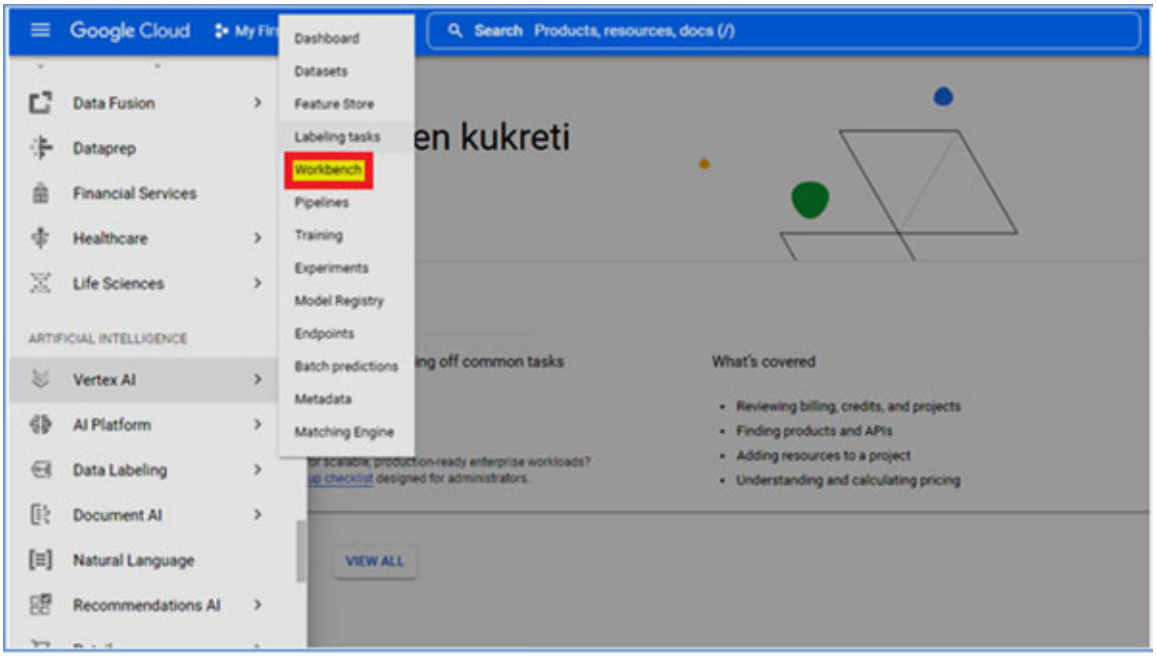

Figure 8.8: *Select vertex AI*

2. If Notebooks API is not enabled, please enable it. Now in Workbench, click on **New Notebook**. Select **TensorFlow Enterprise 2.8 (with LTS)>Without GPUs**. Refer to *figure 8.9*:

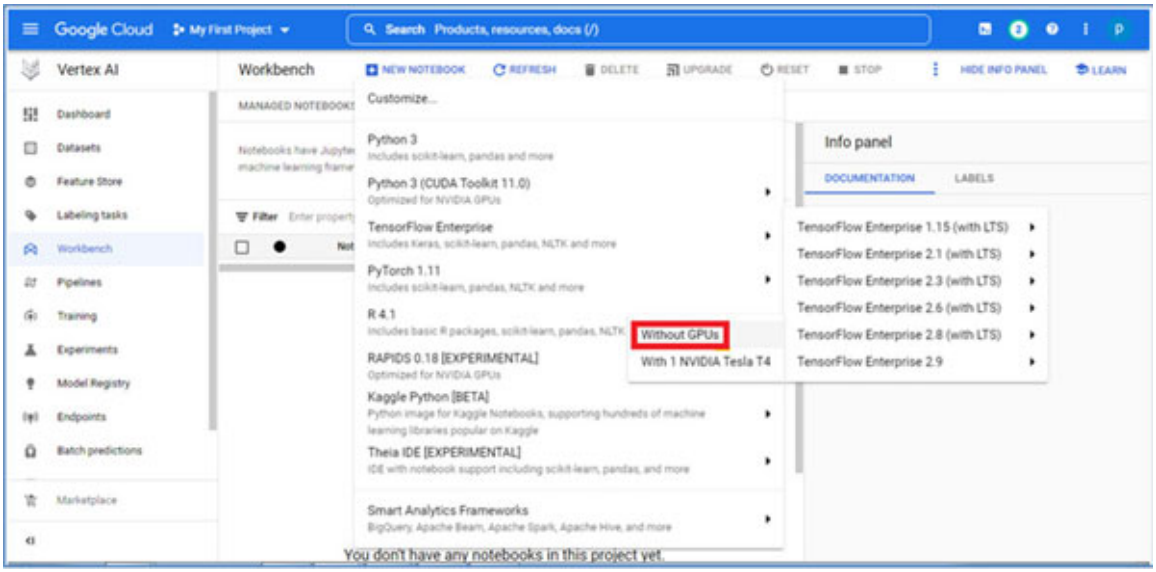

Figure 8.9: *Create new notebook*

3. Select the default name for the notebook or give it a specific unique name of your choice. The select region as **us-west1** and the zone as **us-west1-b**. In the **Notebook properties**, click the pencil icon to edit the instance properties. Refer to *figure 8.10*:

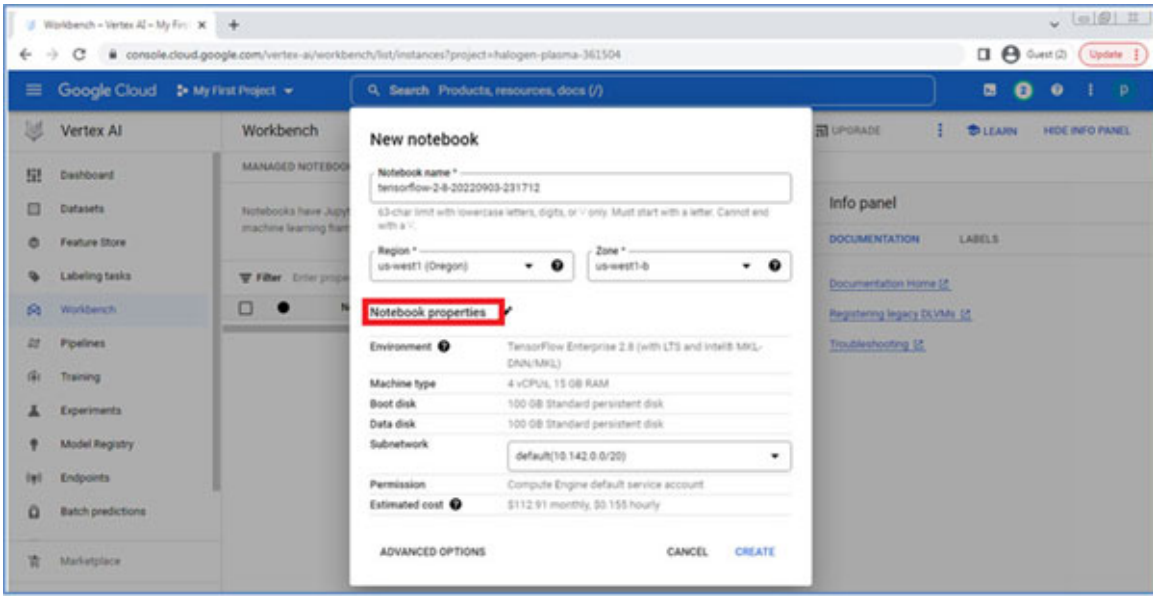

Figure 8.10: *Name the new notebook*

4. Select **Machine type** under Machine configuration as **n1-standard-2**. Leave the remaining values as it is and click on create. Refer to *figure 8.11*:

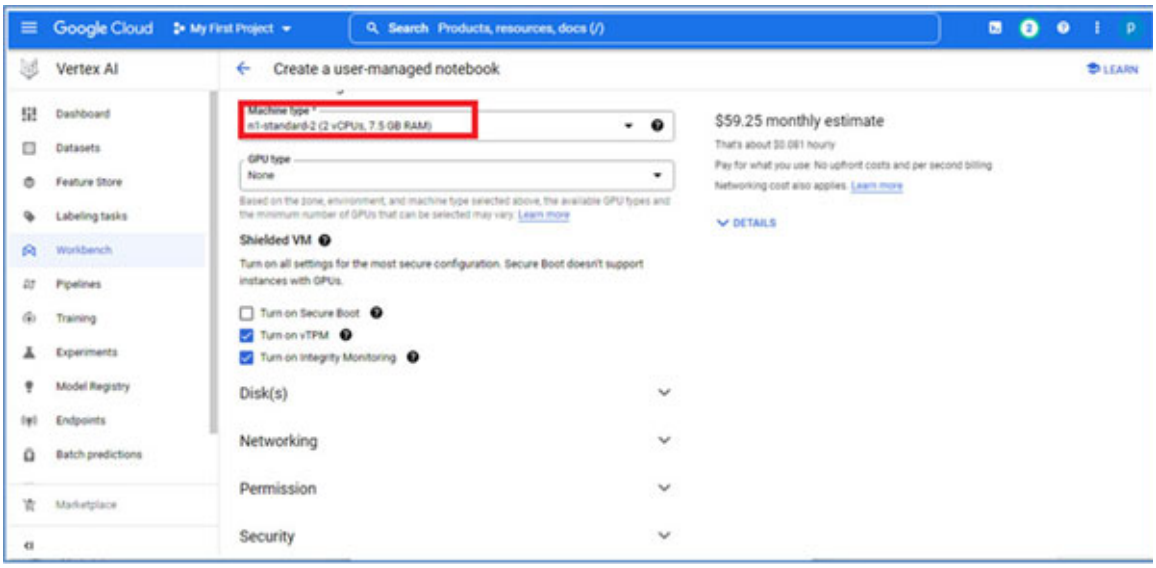

Figure 8.11: *Select machine configuration*

5. After waiting for a few minutes, you will be able to see your instance listed under Workbench. Click on **`OPEN JUPYTERLAB`**. Refer to *figure 8.12*:

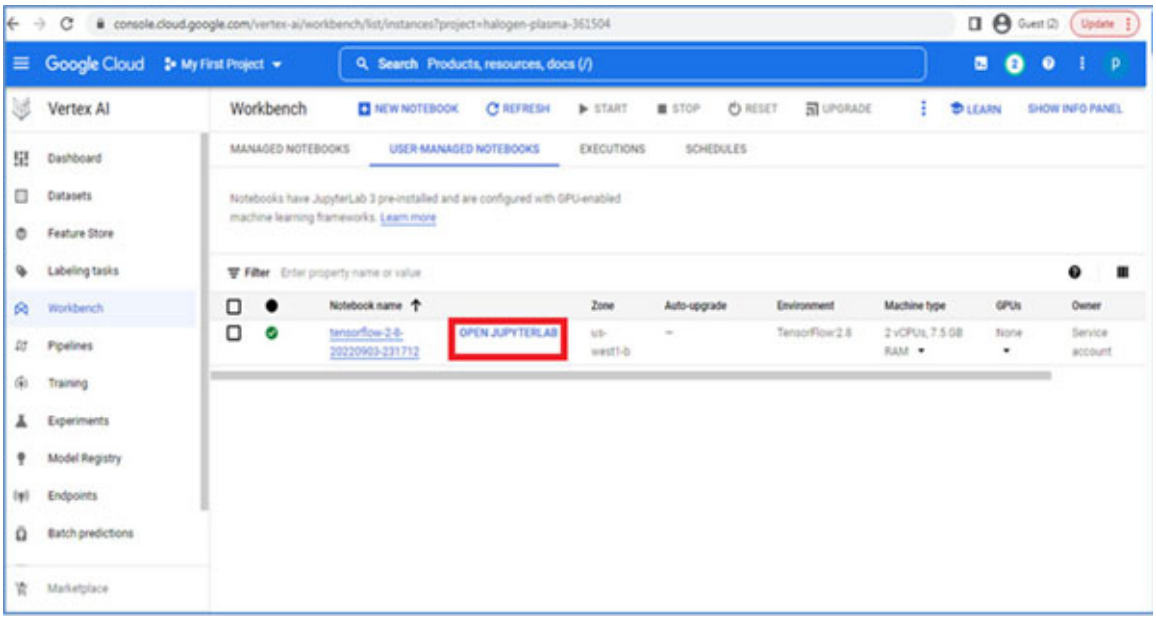

Figure 8.12: *Open instance*

6. Now, we will try to clone the example repo within our Vertex Notebook instance. In Jupyter Lab, click the **Terminal icon** to open a new terminal. Refer to *figure 8.13*:

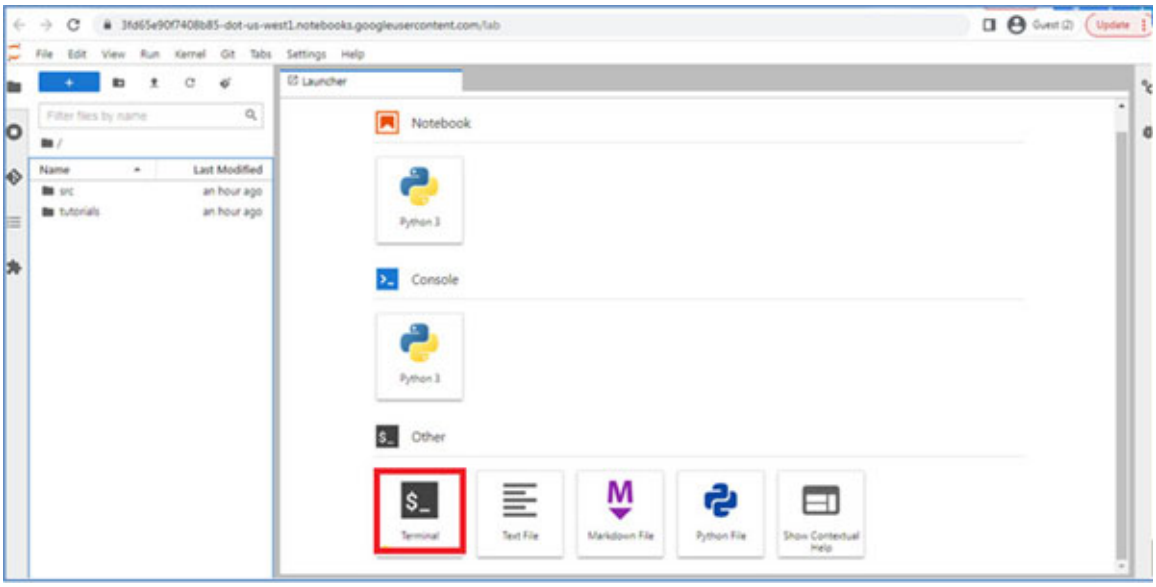

Figure 8.13: *Open a new terminal*

7. In the command line, type the following command to clone **training-data-analyst**. Refer to *figure 8.14*:

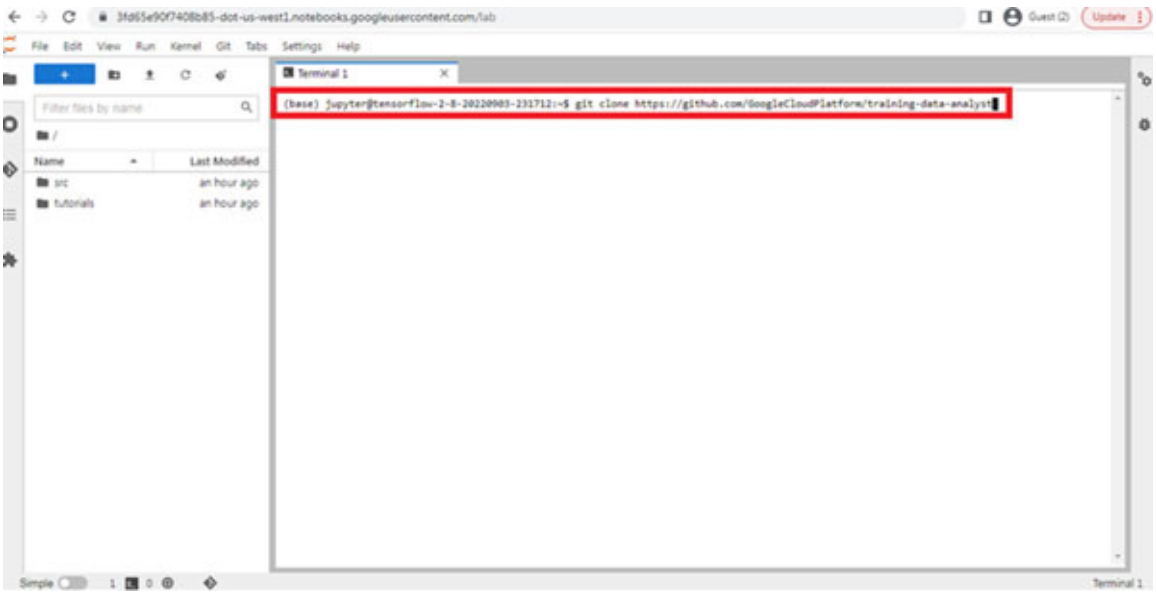

Figure 8.14: *Enter the command*

8. Now, you should be able to see a `training-data-analyst` folder in the left panel of the screen. Click on the folder to see its content. Refer to

figure 8.15:

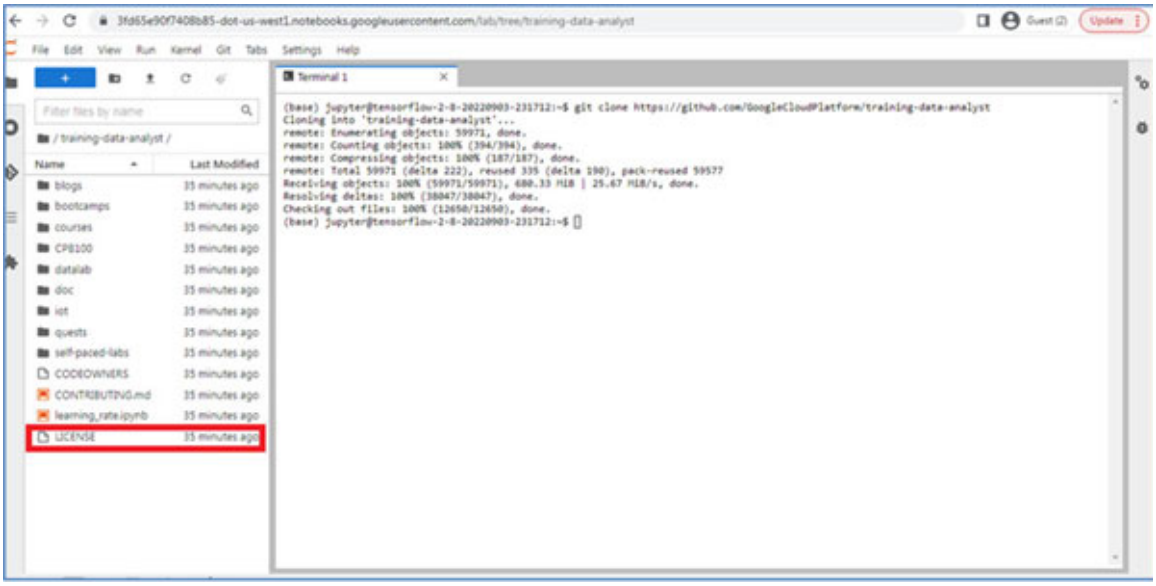

Figure 8.15: *Visible folder in the left panel*

9. Now under **`training-data-analyst`**, click on **`self-paced-labs`** to access the example **notebook**. Refer to *figure 8.16*:

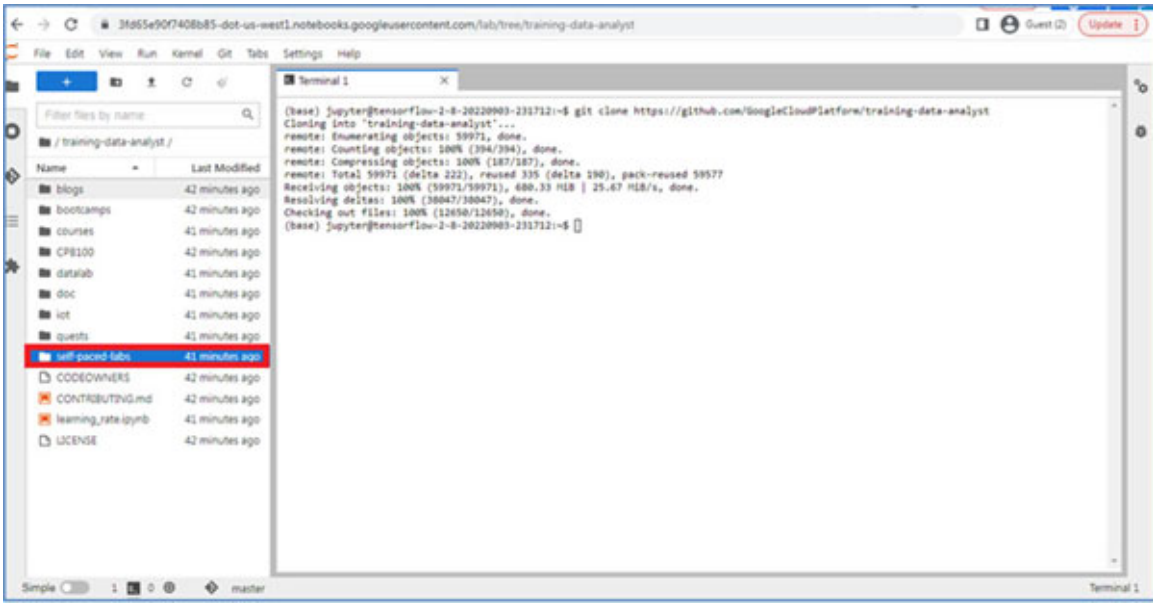

Figure 8.16: *Access example notebook*

10. Now, click on **`ai-platform-qwikstart`** and then open **`ai_platform_qwik_start.ipynb`**. Refer to *figure 8.17*:

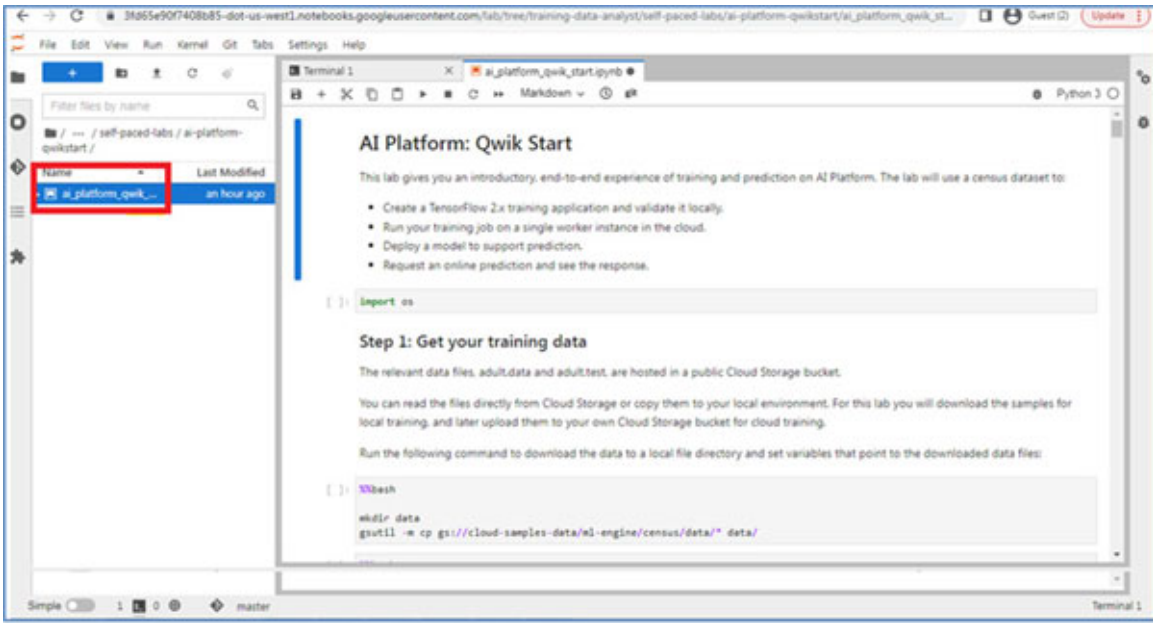

Figure 8.17: *Open ai_platform_qwik_start.ipynb*

11. On the notebook toolbar, clear all output by clicking on **`Edit`** and followed by **`Clear All Output`**. Refer to *figure 8.18*:

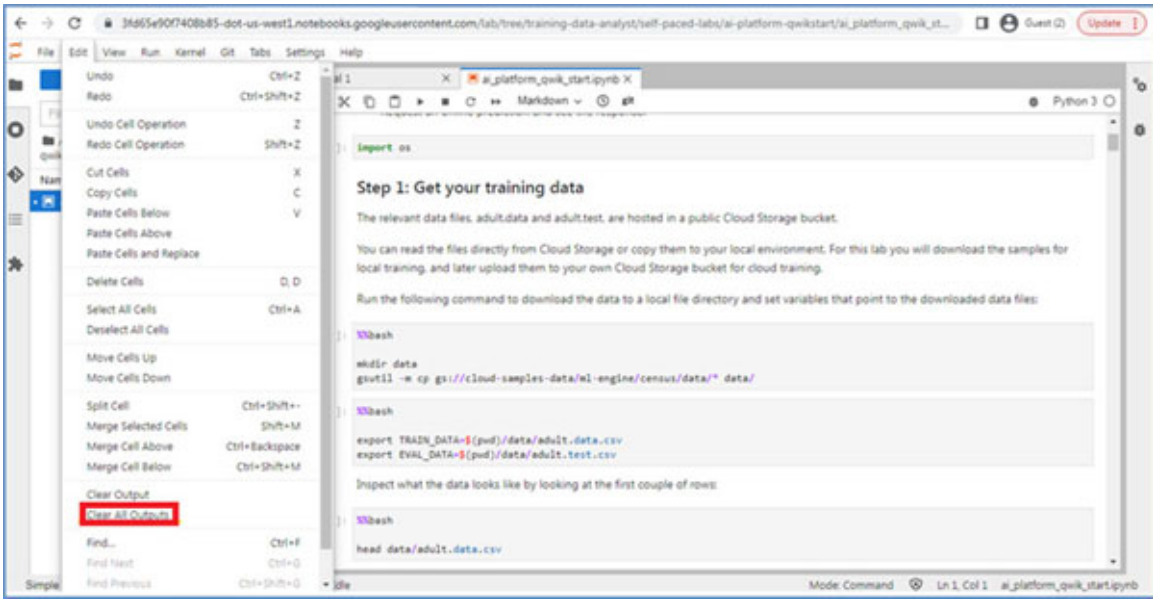

Figure 8.18: *Clear all output*

12. Now you can run your training job by following the instruction in the notebook. Run each step at a time carefully. In these steps, you are importing training data to the cloud storage bucket, training your Machine Learning model using this training data, deploying your training model, and finally, using this training model for online prediction.

Conclusion

In this chapter, we have discussed about basics of AI/ML. Apart from this, we have also discussed about various AI/ML services offered by GCP. We have concluded the chapter with a lab example so that the readers are able to visualize how Machine Learning models are deployed in the Google Cloud Platform. AI/ML is very interesting and one of the most in-demand fields. We strongly recommend readers explore some opportunities in this area if this is an area of your interest. Readers are also advised to go through the links shared at the end of this chapter to develop more understanding of various AI/ML services offered by GCP.

Key terms

- **API**: Application programming interface is a way through which two applications talk to each other.
- **TPU**: Tensor processing unit is an application-specific AI accelerator circuit developed by Google for neural network and Machine Learning.
- **GPU**: Graphics processing unit is a specialized processor designed for rendering graphics and images.
- **textBots**: A tool that performs automated conversational tasks.

Questions

1. What is Machine Learning?
2. What is the various Machine Learning approaches?
3. What are various AI services offered by GCP?
4. Explain neural networks.
5. What is TensorFlow? Explain its various features.

Further reading

- **https://cloud.google.com/products/ai**

CHAPTER 9

Orchestration Services in GCP

Introduction

Orchestration can be defined as the automatic configuration, coordination, and execution of computer systems or services. With respect to services, orchestration has two major areas. One is to create schedules to automate the service execution at a specific point in time, and the second one is the automatic coordination and execution of multiple services in order to achieve the desired results. In data engineering, orchestration refers to the coordination and execution of various ETL/ELT services in some order that are part of some data transformation pipeline. GCP provides various tools and services that can be used for orchestration.

Structure

We are going to cover the following topics in this chapter:

- Cloud Scheduler
- Workflows
- Cloud Composer
- Use cases

Objectives

The objective of this chapter is to familiarize readers with orchestration tools offered by the Google Cloud Platform so that they can make a decision on which orchestration tools or services can be used for a specific requirement. Readers are advised to go through Google's official documentation to develop a deeper understanding of these tools. After studying this unit, you will be able to develop a basic understanding of orchestration and workflow services available in the Google Cloud Platform.

Cloud Scheduler

Cloud Scheduler is a service that is used for scheduling a service execution at a specific date and time. It is an orchestration service offered by GCP that is used for running or executing jobs in GCP, for example, for any batch processing requirement, infrastructure provisioning, or data engineering job. Behind the scene, using a Cloud Scheduler, the cron job gets scheduled, which runs at a specific point in time. You can use a Cloud Scheduler as a single plane of glass for scheduling and managing automation tasks.

Cloud Scheduler helps to minimize manual intervention by allowing you to schedule the jobs at the same time each day, week, or month. It also lets you configure retries and failures after multiple attempts. Some examples are starting the computing engine at a specific time of the day, Triggering Pub/Sub to send a message. Cloud Scheduler provides you with a command line and user interface to view and schedule your Jobs.

Figure 9.1 shows how the Cloud Scheduler configuration screen in the GCP console:

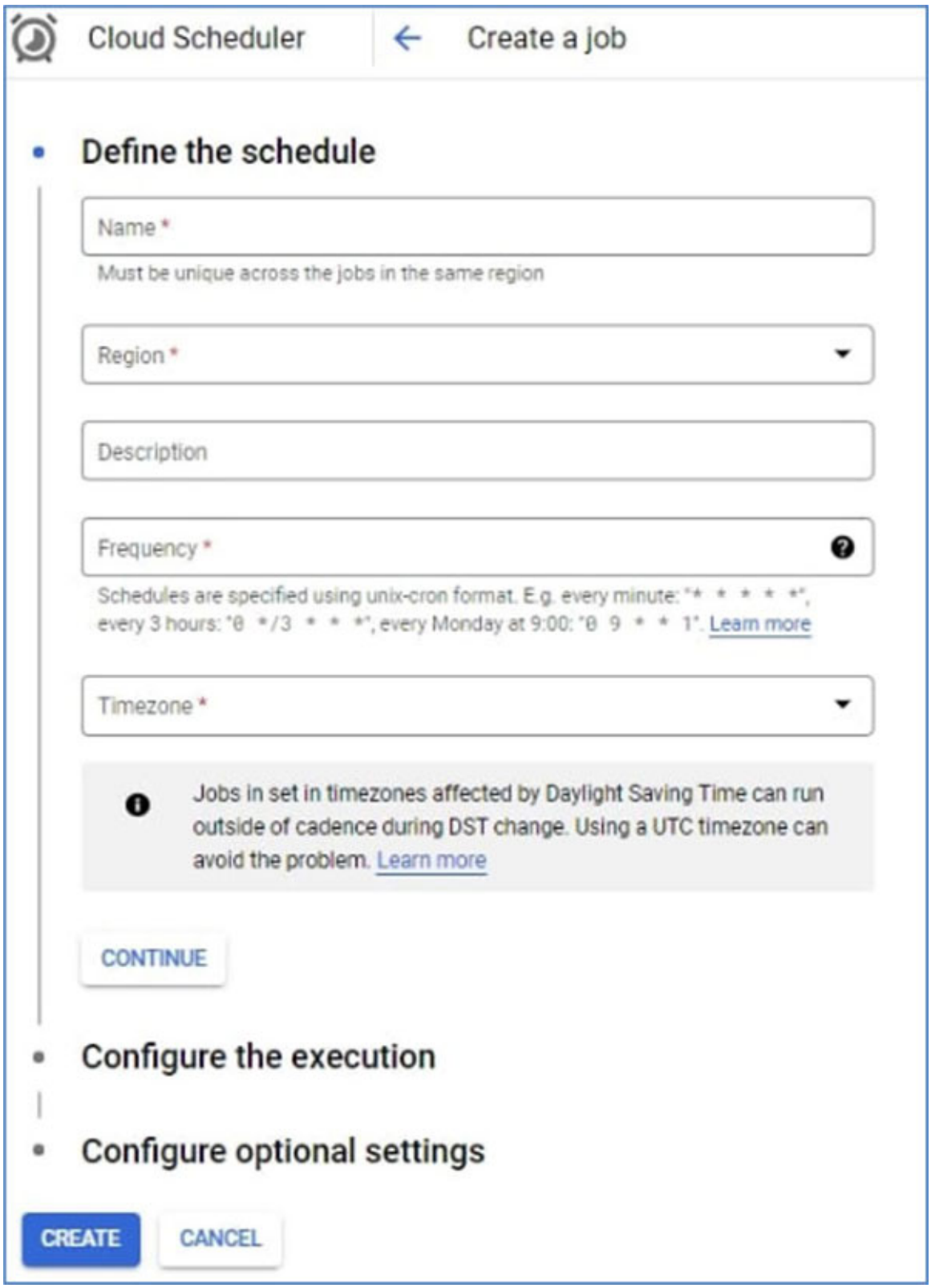

Figure 9.1: *Cloud Scheduler*

Features

Cloud Scheduler has the following main features:

- **Fully managed service**: Cloud Scheduler is a fully managed service. Because it is a fully managed service, users do not require to manage the underlying infrastructure.
- **Support for multiple GCP services**: Cloud Scheduler supports three target types, namely, App Engine, Cloud Pub/Sub, and arbitrary HTTP endpoints. This allows Cloud Scheduler jobs to trigger Compute Engine, Google Kubernetes Engine, Cloud Run, and even On-prem resources.
- **UI and command line access**: Cloud Scheduler provides you centralized management through the command line and UI tools.
- **Cloud Logging integration**: It comes integrated with Cloud Logging that helps in monitoring the running jobs.
- **Guaranteed delivery**: Cloud Scheduler guarantees at least one delivery to the target service.

As of date, Google allows you to run three jobs in a month for free, and post that it will cost you 0.10$ per month.

Workflows

Cloud Workflows is a serverless orchestration platforms like Scheduler; however, in Workflows, multiple HTTP-based services running on Cloud Functions or Cloud Run are chained together in the form of a stateful workflow. Workflows are useful in chaining multiple microservices together, doing API integration with external systems, or infrastructure provisioning in the cloud platform.

Features

The following are some of the important features of Workflows:

- **Fast deployment**: Google Cloud Workflow is fast to deploy.
- **Fault-tolerant**: Workflow gets deployed to multiple zones by default. Hence, it is fault tolerant.
- **Secure**: Workflows run in a sandbox environment; hence, they are secure.

- **Low latency**: It provides faster execution between the steps; hence, overall latency is low.
- **Multiple ways to trigger**: Workflows can trigger manually, through some API call, or can be triggered through the program.
- **Integration with cloud logging**: Workflows come integrated out of the box and also come integrated with Cloud Monitoring. Therefore, you can get insights into Workflows when they get executed.

Workflows are defined using YAML or JSON, which is also called definition. Output from one step is fed as input to another step. Once a definition file is created, we need to deploy and execute it to produce the desired results. Since Workflows are serverless, they can scale out rapidly to perform multiple executions at the same time. You can execute Workflows using Google Cloud CLI, Google Cloud Console, or Restful API.

While defining, you can control its execution by utilizing steps, conditions, iterations, and sub-Workflows:

- **Steps**: Steps are a series of tasks that you are going to perform in order. The output of one step can work as an input for another step.
- **Parallel steps**: A parallel step is used to define a part of the Workflow where two or more steps can execute concurrently.
- **Conditions**: Using conditions, you can control the flow of execution for a Workflow. You can use a switch block to control the Workflow execution.
- **Iteration**: You can run a "for loop" to perform an iteration while executing a workflow.
- **SubWorkflows**: When you want to encapsulate multiple steps and iterate them multiple times, you can make use of SubWorkflows. SubWorkflows are equivalent to a function in a programming language.

You can trigger the execution of Workflows in various ways. You can trigger them manually, programmatically, through scheduling, or by adding them to the environment runtime.

Figure 9.2 features a Workflow structure:

Figure 9.2: *Workflow structure*
Source: *https://cloud.google.com/Workflows/docs/reference/syntax/steps*

Cloud Composer

Cloud Composer is a fully managed Workflow orchestration service that is built on Apache Airflow. It is used to create data-driven Workflows. As it is a fully managed service, you need not to bother for managing, maintaining, or configuring it. In this, you create tasks associated with each step and configure the order of those steps. These tasks are usually written in python and are represented by **Direct Acyclic Graphs (DAGs)**. DAGs are Workflows that have multiple tasks which are supposed to get executed in some order as defined by their relationships and dependencies. It helps to build and execute a data pipeline.

In the chapter Big Data in Google cloud, we have already discussed about various data ingestion, transformation, and analytics services in Google Cloud and their use cases. While creating a data transformation pipeline, we make use of different services for different purposes. For example, we may require creating a data pipeline for streaming data that needs to go through ingestion, transformation, loading, and analysis. In DAGs, you can define these tasks in the form of some steps and define a workflow orchestration to get these tasks executed in a specific order. Cloud Composer runs on a GKE cluster and uses Cloud Storage Buckets to store DAGs, logs, and various plugins.

Figure 9.3 features a Cloud Composer environment:

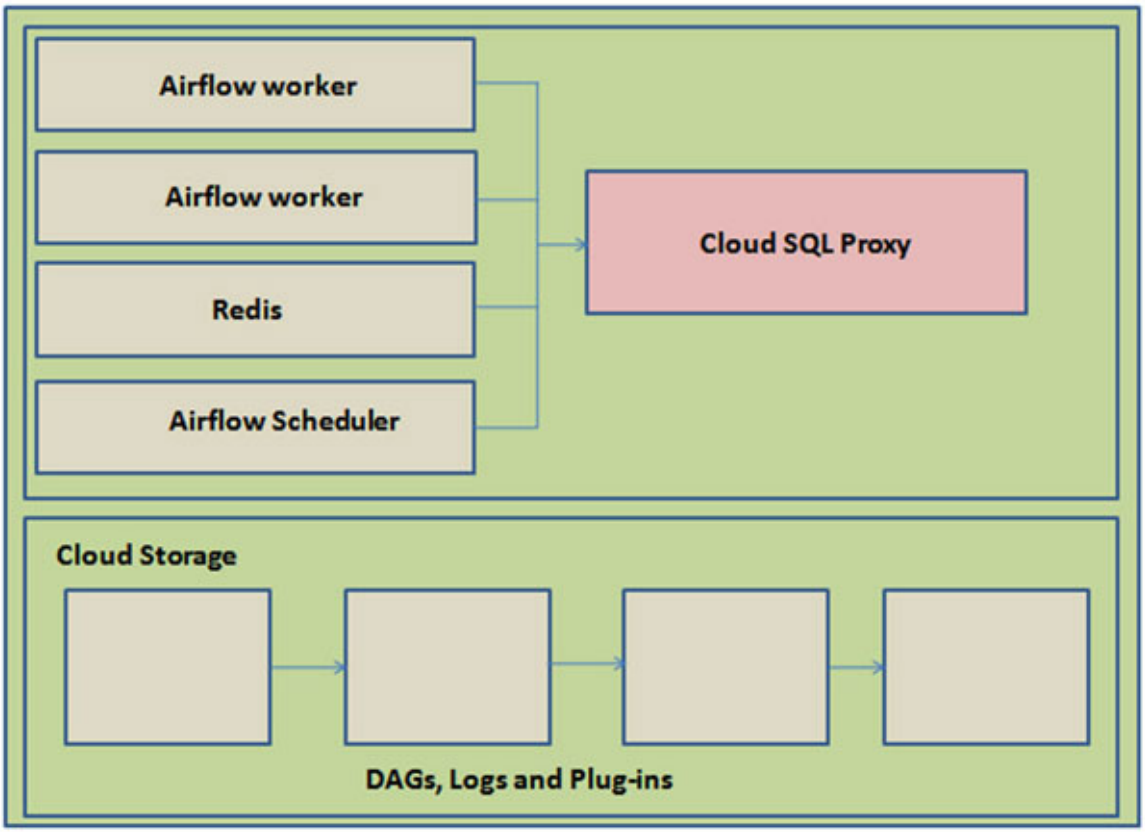

Figure 9.3: *Cloud Composer environment*

Features

The following are some of the important features of Cloud Composer:

- **Fully managed**: Cloud Composer is a fully managed service thatlets you focus on orchestration rather than resource provisioning.
- **Multicloud support**: Using Cloud Composer, you can create Workflows across multiple clouds.
- **Open Source**: Cloud Composer is an open-source offering that builds upon Apache Airflow. Being open source helps with avoiding any vendor locking.
- **Integration**: Cloud Composer has built-in integration with many GCP services such as BigQuery, Dataflow, Dataproc, and so on.
- **Support for Hybrid cloud**: Using Cloud Composer, you can orchestrate Workflows between On-Prem and cloud environment.

Selecting the right orchestration service in GCP

Figure 9.4 is a decision tree that can help you in identifying the right orchestration service for your requirement:

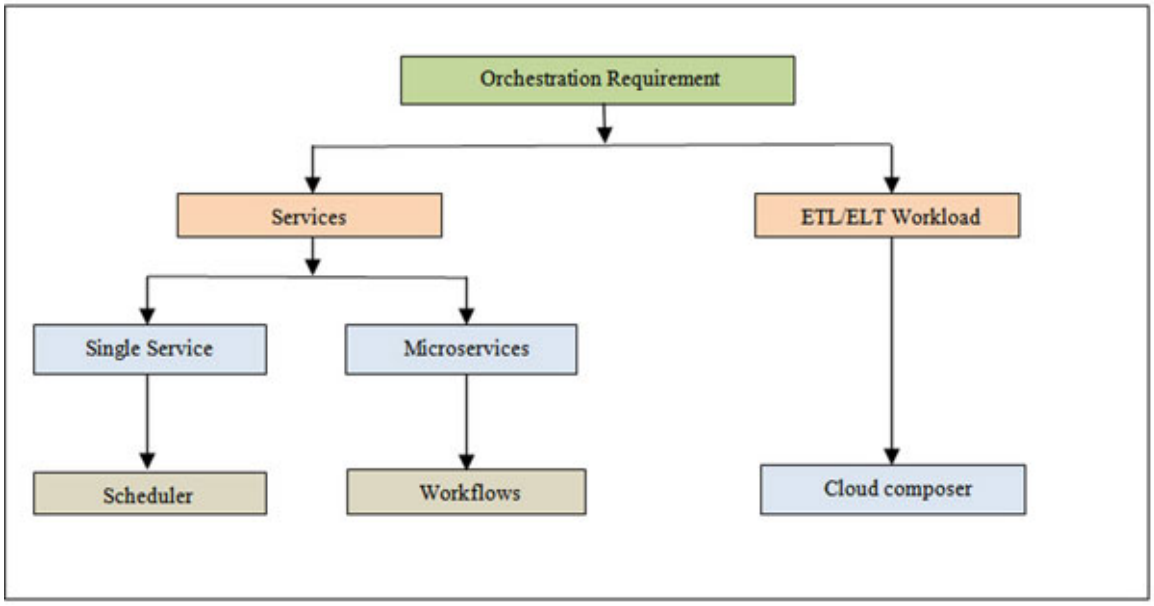

Figure 9.4: *Decision tree*

Use cases

Now, we will try to explore a few use cases for Cloud Scheduler, Workflows, and Cloud Composer, respectively.

Cloud Scheduler use cases

The following are some of the use cases for Cloud Scheduler:

- **Infrastructure provisioning using Cloud Scheduler**: There are scenarios where you want your VM running only on specific days of the week or may require your VM to be shut down during the night. Cloud Scheduler can automate these tasks for you and save your efforts.
- **Scheduling your Big Data jobs**: Cloud Scheduler can help you with running recurring Big Data jobs. For example, you can run a Cloud Function that takes inputs from the Cloud Pub / Sub messaging service every 15 minutes and writes this output to any target service.

- **Triggering App Engine**: You can trigger your application running on the App Engine at a specific time on any specific day as per your requirement.

Workflows use cases

The following are some of the important use cases for Workflows:

- **Microservices orchestration**: Microservices orchestration is one of the most important use cases of Workflows. You can invoke a series of microservices in a specific sequence while running an application using Workflows.
- **API integration**: You can schedule Workflows in specific use cases where you need to invoke some API to perform any task. For example, you can create a Workflow in which, once you upload any document to a Storage Bucket, it invokes the document AI. Once the document inspection is complete, a Cloud Function can be triggered to provide the required approval.
- **Infrastructure provisioning**: You can create Workflows to orchestrate the infrastructure provisioning steps.
- **Data pipelining requirements**: You can orchestrate data pipelines using Workflows. For example, for batch or stream processing requirements, you can orchestrate a sequence of steps to perform ingestion, transformation, and analysis using various GCP services. Workflows provide built-in connectors for these services.

Cloud Composer use cases

The following are some of the important use cases for Cloud Composer service:

- **Data pipeline orchestration**: Cloud Composer helps you in orchestrating various services that are part of data transformation pipelines. For example, for any stream processing requirement of any IoT data that needs to be ingested through messaging service like Cloud Pub/Sub, transformed through some service like **Dataproc** or **Dataflow**, and presented to a warehouse service like **Google BigQuery**, you can make use of Cloud Composer service to do the required orchestration for

you. You can write DAGs to define your orchestration Workflow for various services.

In *figure 9.5*, you can see a high-level overview of the data pipeline use case:

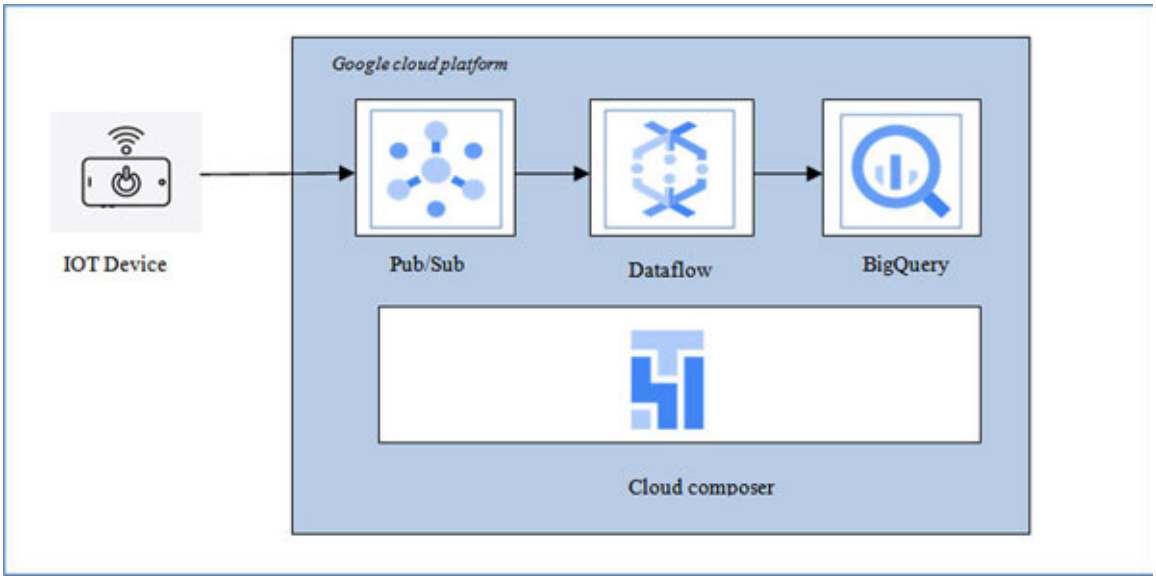

Figure 9.5: *Decision tree*

- **Infrastructure provisioning**: Cloud Composer can help with orchestrating complex infrastructure provisioning tasks. One of the use cases can be if you want to take snapshots of multiple instances running on a different environment, you can orchestrate activities like stopping these instances at a specific point in time depending upon schedule, taking snapshots of these instances, and restarting these instances after taking snapshots.

Conclusion

In this chapter, we have discussed about various orchestration services available in the Google Cloud Platform. So far, we have understood that there are various orchestration services that can solve interesting use cases. Orchestration services help with reducing the overhead of executing tasks in a sequence and also provide required connectors that help in integrating with various GCP services. We have tried touching on the basics of these services, considering the scope of this book. Readers are advised to go through the GCP site and other online study material to develop a further understanding of this.

In the upcoming chapter, we will discuss about migration services used in the Google Cloud Platform.

Key terms

- **DAG**: DAG is a collection of all the tasks you want to run in the form of a workflow, organized in a way that reflects their relationships and dependencies.
- **Microservices**: Small loosely coupled services that are part of an application.

Questions

1. What do you understand by Orchestration?
2. What are various orchestration and scheduling services available in GCP?
3. Please explain some of the features of the Cloud Composer service available in the Google cloud platform.
4. Please mention some of the use cases of Scheduler, Workflows, and Cloud Composer services.

Further readings

- **https://cloud.google.com/scheduler**
- **https://cloud.google.com/workflows**
- **https://cloud.google.com/composer**

CHAPTER 10

Migration Services in GCP

Application migration is the process of moving a software application from one computing environment to another. You might, for instance, migrate an application from one data center to another, from an on-premises server to a cloud provider's environment, or from the public cloud to a private cloud environment.

- IBM

Introduction

The above mentioned statement clearly defines some requirements for migrating applications. Apart from moving some applications or a set of applications to the cloud, there can be a requirement of moving an entire data center to the cloud. Some of the primary reasons why organizations are looking to migrate their workload to the cloud are less management overhead, cost optimization, faster innovation, and so on.

Structure

In this chapter, we will cover the following topics:

- Migration concepts
- StratoZone
- Migrate to Virtual Machines
- Transfer Appliance
- Storage Transfer Service
- Database Migration Service
- BigQuery Data Transfer Service
- Lab activity

Objectives

The objective of this chapter is to familiarize readers with various migration services available in the **Google CloudPlatform** (**GCP**). GCP provides a number of tools and services for various kinds of migration requirements. These services are well designed for achieving the required results, with very less to no management overhead. This chapter will help the readers to understand the basics of these tools or services. Readers are advised to go through the official Google documentation for gaining more insight.

Migration concepts

Migration is a broad topic that can be learned through a combination of theory and practice. In this section, we will try to understand the various concepts in migration.

Migration can be divided into the following four phases.

Discovery

This is the first and one of the most important steps in migration. Before migrating an application or an entire data center, we need to understand the existing environment(s), its applications, and their associated infrastructure. This is necessary to design the target environment(s) in the cloud. There are various discovery tools available that can run on the existing environment to collect data from the source environment. Some examples of discovery tools are BMC ADDM, StratoZone, Dynatrace, and so on. Organizations usually maintain a **Configuration Management Database**(**CMDB**), which contains an inventory of the existing environment. It also contains all the information about hardware and software components and the relationship between these components. However, sometimes data collected through CMDB is incomplete or does not contain all the required information. Therefore, organizations prefer running some discovery tool to gather this configuration data.

Figure 10.1 features a Configuration Management Database:

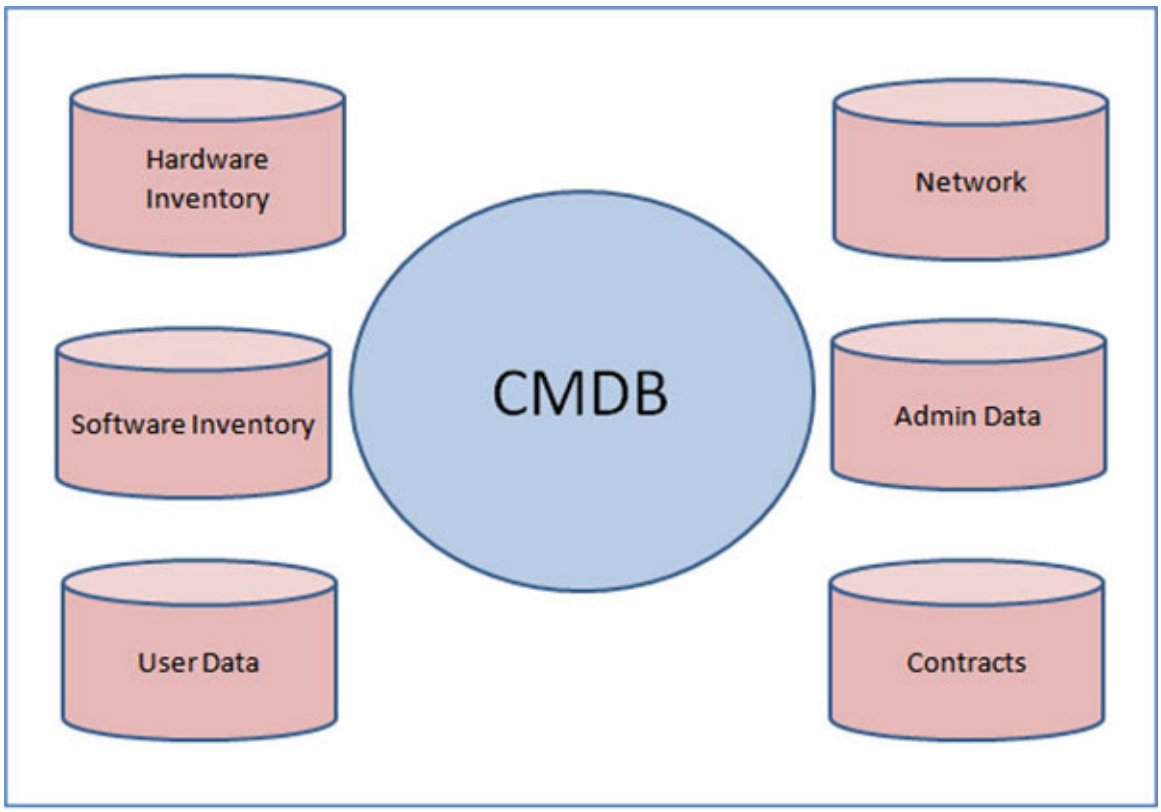

Figure 10.1: *Configuration management database*

Assessment and planning

After discovery, the next important step in the migration journey is assessment. We need to understand how different applications are dependent on each other, what platform they are running on, what are the compute, memory, storage, and database requirements we have, and what are the different environments (production, development, test, and so on) in the source side. Understanding API integration is also part of the assessment. The planning phase involves defining migration strategies. After the assessment, the migration team does the required planning for the actual migration. RLANE is an important part of any application migration strategy. RLANE defines various migration strategies from an applications point of view that require a thorough analysis of the applications and their dependencies. These strategies are primarily used to define the migration approach for moving applications or applications to the cloud.

The following are the five main RLANE strategies:

- **Rehost**: The Rehost strategy is followed when we want to perform lift and shift migration of an application or applications to the target environment. Since we are mainly focusing on cloud-specific migrations here, Rehost can be referred to as moving an application to the target cloud environment without making any changes to the existing application. Usually, it is easier to make any architectural changes to an application once it has already moved to the cloud than migrating after making the required changes. Rehost is still one of the most widely used strategies for migration, considering its quick implementation.
- **Replatform**: Replatform strategy is followed to make use of cloud-native services. In this approach, you do not make any architectural changes in the applications. However, you make use of cloud-native services such as managed database services or any other serverless offering. This reduces the management overhead and may also result in license optimization in some cases.
- **Refactoring**: Refactoring requires some minor changes in the application code before it gets moved to the target environment. For example, there may be some static values in the application code (such as IP address or MAC address) that may get changed once the application moves to the target environment. In this, we are not redesigning the application from scratch, and no changes are made to the application functionality.
- **Rearchitect**: As the name suggests, re-architecting is the process of redesigning the architecture of any application to gain certain benefits. For example, you may require rewriting the application code in order to improve its performance, or you may think of converting any monolithic application into a microservices-based architecture, which may require you to rewrite your application code from scratch. Application rearchitecting is a time-consuming activity. Before finalizing this strategy, we need to make a proper estimation of the time that we may require to rearchitect the application and compare it with the time that we have in hand to perform the migration. A more practical approach is to rehost the application to the cloud and then rearchitect the application.
- **Retire**: There may be some applications that are not being used at all. After doing a thorough assessment, you may decide to retire those applications. It has been observed that after doing planning and assessment, around10% of applications fall into this category.

- **Retain**: Retain, as the name suggests, means that we do not migrate an application at this stage and retain it in the existing environment. There can be various reasons behind it. For example, an application can be a legacy application, and moving to the cloud can impact its functionality. Other examples can be that the application has a major refactoring/rearchitecting requirement before it can be moved to the cloud, and due to time constraints, we decide not to move it to the cloud.
- **Replace**: After doing a thorough assessment of the application, we came to the conclusion that the application cannot be moved to the cloud, and hence, we decided to retire the application and replace it with a new one.

Once you have categorized applications into one of the preceding categories, the next step is to decide which applications are to be moved first. The usual approach is to move the less complex applications that have fewer dependencies on other applications, and therefore, take less time to move. Also, at first, we try to move the applications that are in the non-production environment. This approach helps us to test our migration with minimal risk. This concept of testing the migration by moving some specific applications is called **Proof of Concept**(**POC**).

Another point that needs to be considered while planning the migration is to consider the application tiers. We can first think of moving the Database tier, followed by the Application tier and Web tier. We need to make sure that network and shared services (that is, Active Directory, exchange, and so on) are in place before moving an application to the cloud. We usually categorize the migration in different waves after considering bandwidth, timeline, and application dependency requirements. This requires a lot of brainstorming and discussions between migration teams and application SMEs.

Migration

After discovery and the assessment and planning phases are complete, we proceed with the actual migration. By this time, we definitely have a set of applications that are supposed to get migrated. We are also supposed to have phase-wise migration details. We make sure that the cloud foundation or landing zone in the target environment is also in place. In a Landing zone, we have organization structure, network configuration, security configuration, and access permissions already defined. It is a foundation that you build to create and deploy your cloud resources. Source to target connectivity should be established and it should fulfill bandwidth requirement to complete migration

within the defined timeline. As discussed earlier, migration usually starts with a POC. A successful POC is required to build the confidence among all the stakeholders, primarily of the customer and the executor. Once all the applications that are part of POC have migrated successfully, we proceed with other migration waves. These waves are already defined during the planning phase.

Testing, documentation, and knowledge transfer

Once migrations have been completed successfully, the applications need to get tested against any latency or accessibility issues. During various phases of migration, various teams that are involved perform the necessary documentation. In the end, all the required documents that have information about the target designs and other necessary details are handed over to the operations team, which is responsible for managing applications and the target infrastructure. Successful testing of the applications and knowledge transfer concludes the overall migration activity. *Figure 10.2* features the various phases during migration:

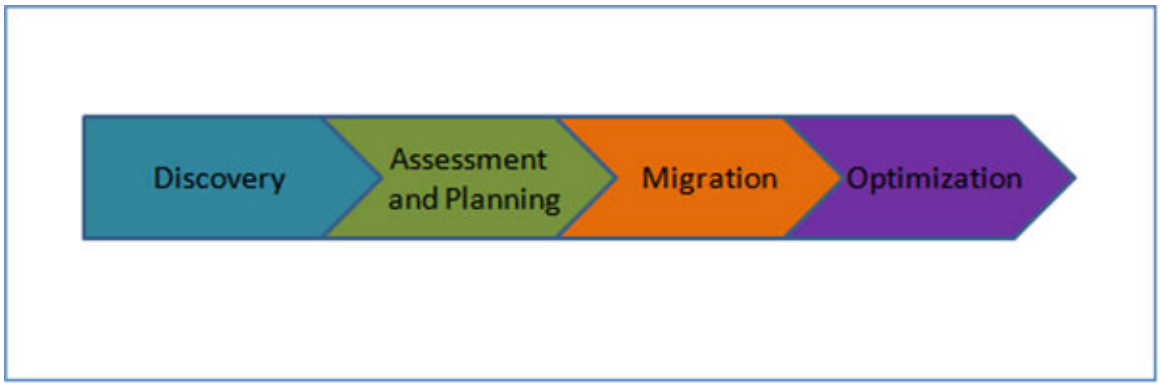

Figure 10.2: *Various phases during migration*

Note: Optimization is continuous improvement in the target environment during operational activity.

StratoZone

GCP provides a StratoZone tool for discovery and assessment. This tool can be used to perform the discovery of any On-Prem, Google Cloud, or any third-party cloud service provider. This tool automatically discovers the existing

infrastructure, does the required analysis, gives recommendations, and hence, helps with planning the migration.

StratoZone has two major components.

StratoProbe

StratoProbe is a data collection software that is installed in a Windows machine and collects required metrics from servers and databases in the customer's environment. These metrics are then used to do the migration planning. You can perform discovery of the resources by manually entering specific IP addresses of the target systems, or you can provide a range of IP addresses. **StratoProbe** can discover the assets within that IP range and collect the data. This data is then sent to the **StratoZone Portal**. StratoProbe can only collect data from Windows and Linux-based systems and can scan only MySQL, MSSQL, PostgreSQL, and Oracle databases. StratoProbe collects the following data from the source systems:

- Machine-level data
- Configuration data
- Utilization data
- Network dependencies
- Installed software

StratoZone portal

StratoZone portal allows you to perform assessments that are needed to assess the source environment using the metrics. You create credentials in the portal that are required for running StratoProbe in your source environment. Once the StratoProbe has collected the required metrics from the source systems, it sends these metrics to the StratoZone portal. StratoZone portal does the required assessment of the collected data and does the cost comparison with the cloud-recommended services.

Figure 10.3 shows the StratoZone Portal:

Figure 10.3: *StratoZone portal*
Source*: https://cloud.google.com/migrate/stratozone/docs/stratoprobe-verify-results*

Note: There are some scripts that you can install and run-on other cloud providers such as AWS and Azure. These scripts can be used to collect important performance and usage data. This collected data can then be imported into the StratoZone portal to perform the required assessment and provide recommendations.

Migrate to Virtual Machine(V5.0)

Migrate to virtual machine (Previously called **migrate for compute engine**) is an offering from GCP which is used to move your virtual machine from the On-Prem data center to compute engine in GCP. Migrate to virtual machine uses replication technology to migrate the **virtual machines** (**VM**) residing in the On-Prem system to GCP.

Pre-requisite for performing migration using Migrate to Virtual Machine

The list of pre-requisites that are needed before performing a migration using **Migrate to Virtual Machine** is as follows:

Installing migrate connector: Migrate connector should be installed in the On-Prem data center. Migrate connector performs many important tasks, such as establishing a secure connection between the On-Prem data center and Google Cloud API over port 443, performing storage operations on VM disks using vSphere APIs, and executing necessary tasks for stopping and monitoring On-Prem VM using vSphere APIs.

Connectivity between On-prem data center and GCP: There should be connectivity between the On-prem data center and Google Cloud Platform. You can establish this connectivity using a Dedicated Interconnect, Partner Interconnect connection, or using VPN over the public internet.

Creating Landing zone in Google Cloud Platform: You need to setup a Landing zone in GCP, which should have the host project in place. This project can work as a target project, and moreover, this is the project where you enable Migrate for Compute Engine API. You also need to have appropriate Compute Engine resources configured inside the host project that will be used as your migration target. You can also have additional target projects that you have access to, apart from this host project, and use Compute Engine resources in these target projects as your migration targets. Apart from this, you also need IAM permissions in place to access the Compute Engine resources.

Figure 10.4 features a Migrate for compute engine example.

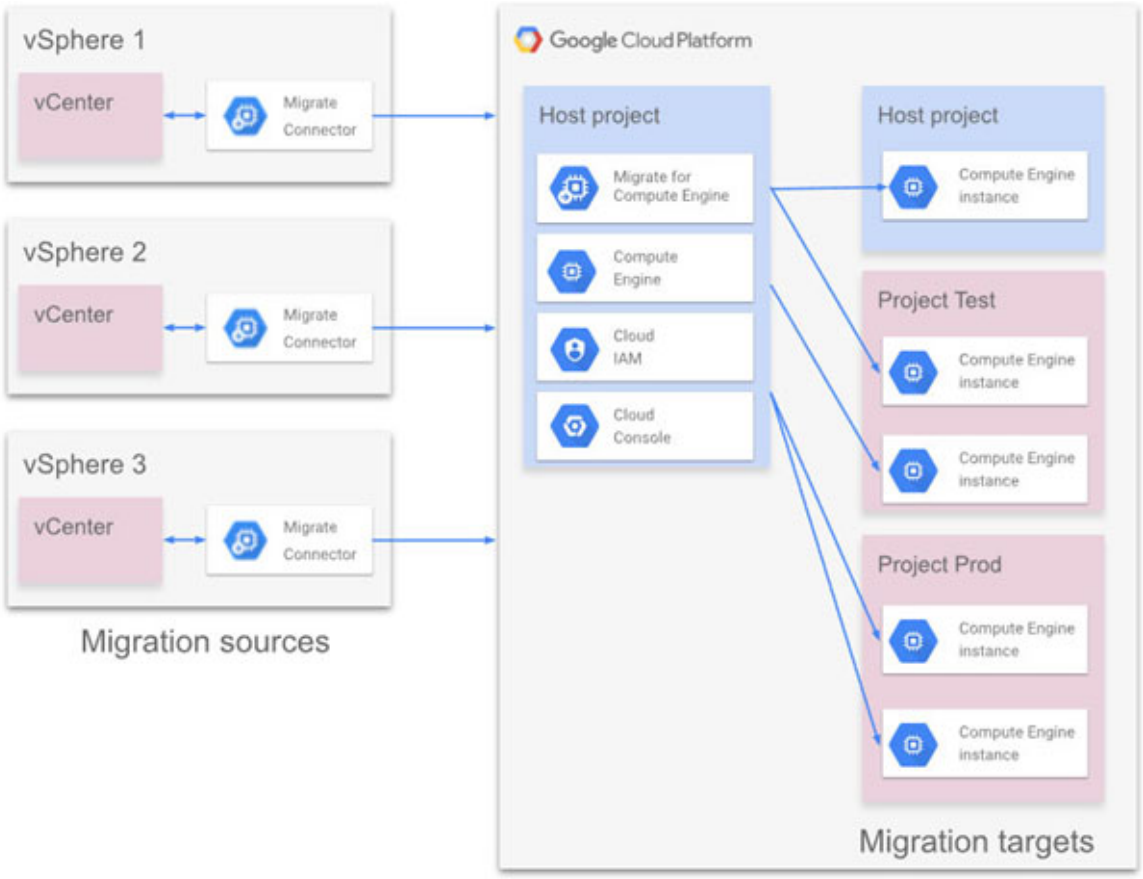

Figure 10.4: *Migrate for virtual machines example*
Source*: https://cloud.google.com/migrate/virtual-machines/docs/5.0/concepts/architecture*

Various phases in the migration lifecycle

The main phases involved in the migration process are as follows:

- The first phase includes selecting one or multiple sources VMs, that need to get migrated to GCP. This phase is called **on-boarding**.
- The next step is **replicating** data from the source VM to the target GCP environment. This is a continuous replication. During the first time, it is a full replication. Once full replication is complete, there will be incremental replications from source to target before performing a final cutover.
- Select project, target Compute Engine VMs, target Compute Engine VMs with required memory, compute, and so on occurs during this third phase.
- You also get an option to create a **test-clone** Compute Engine, which can be tested before moving to the actual Compute Engine VMs.

- Finally, shut down the primary or the source VM and perform a final sync between the source and target. This phase is called the **cutover** phase. Once this activity is complete, traffic can get redirected to the target Compute Engine VMs, and final cleanup can be done.

Transfer Appliance

A Transfer Appliance is a device that is used to securely transfer large amounts of data offline from customer's existing data centers facility to the Google Cloud Platform. Transfer Appliance is suitable for conditions when the data that needs to be transferred is very large and cannot be migrated or moved over the wire. The data that needs to be transferred is copied to the appliance and encrypted using industry-standard AES256 encryption before it gets shipped and uploaded to a Google upload facility. Transfer Appliance comes in three sizes: 7 TB, 40 TB, and 300 TB.

Transfer Appliance comes with various security features such as hardware attestation, tamper resistance, encryption, and **Trusted Platform Module** (**TPM**) chip. Transfer Appliance is available at specific locations, so before ordering a Transfer Appliance, the customer needs to check if the appliance is available for your location. Transfer Appliance comes with SSD drives to ensure that the data transfer is fast. You can either use 10 Gbps RJ45 interface or 40 Gbps QSFP interface to transfer your data to the appliance.

Use cases

Some of the important use cases for transfer appliances are as follows:

- **Larger amount of data migration requirement**: Sometimes, the data that you want to transfer is of a large size, and you do not have enough bandwidth to support the transfer within the defined timeline. For example, if you want to transfer 300 TB of data using a 100 Mbps connection, it may take you upto nine months to transfer that data to a Cloud Storage Bucket. However, by using a Transfer Appliance, you can transfer and upload this data to a Google Cloud Bucket within 50 days.
- **Transferring Archive data**: Backup and Archive data is not live data and may require occasional access. As data availability is not a major issue here, you can opt for doing an offline transfer using the Transfer Appliance.

- **Copying redundant data from On-prem to cloud environment:** Sometimes, customers like to perform some experiment on the data that is available On-prem. They can copy this data to the cloud environment and perform some testing on it before moving an actual workload to the cloud.

You can apply for a Transfer Appliance from your Google Cloud Console after logging in to your project. You should have an owner role in your project for applying for an appliance. The appliance comes in online and offline modes. Once the data is copied to the appliance, it is streamed from the appliance to the GCS bucket. It requires outbound internet access.

Figure 10.5 illustrates applying for a transfer appliance.

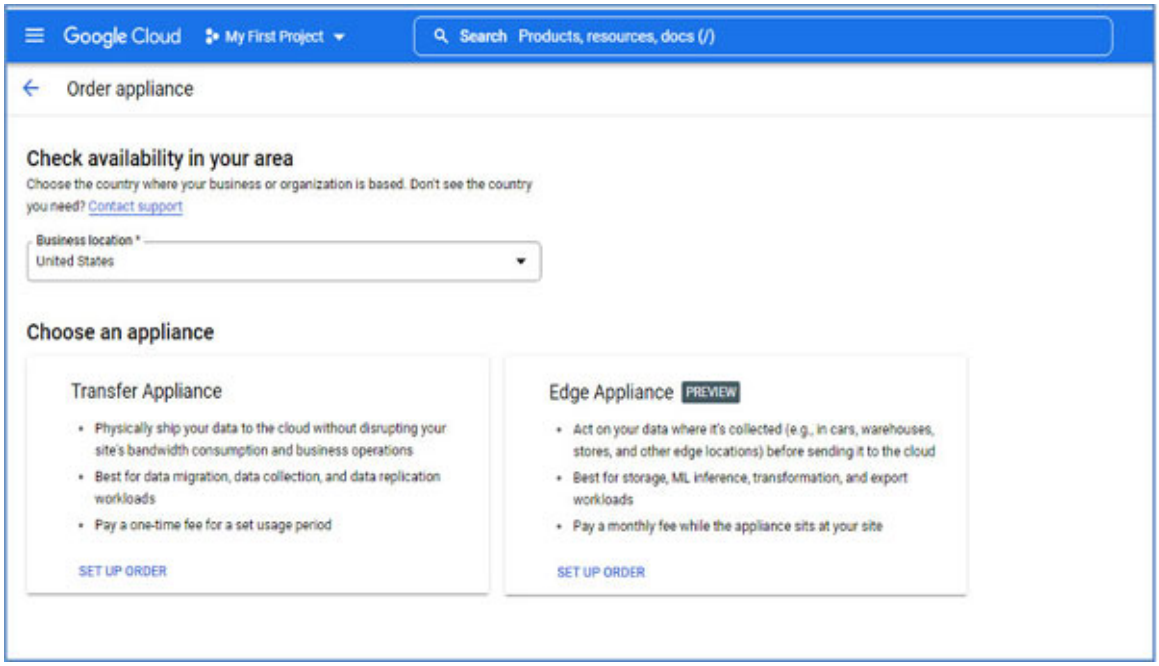

Figure 10.5: *Applying for transfer appliance*

Storage Transfer Service

Storage Transfer Service (STS) is a service offered by Google Cloud Platform that is used to transfer data between objects and file storage across different environments. Source and destination can be on On-Prem, GCP, AWS, Azure, and so on. You can perform the migration using Google Cloud Console, CLI, client libraries (Java and Python), or API. Using STS, you can transfer data between different buckets, from one file system to another file system, or you can also backup data from a Cloud Storage Bucket belonging to

any cloud service provider to GCP Cloud Storage. For transferring data from the On-prem system, you need to install and start the agent in the source system. Once this is done, you can data transfer the Job to the cloud console.

Figure 10.6 shows the Storage Transfer Service and On-prem data transfer:

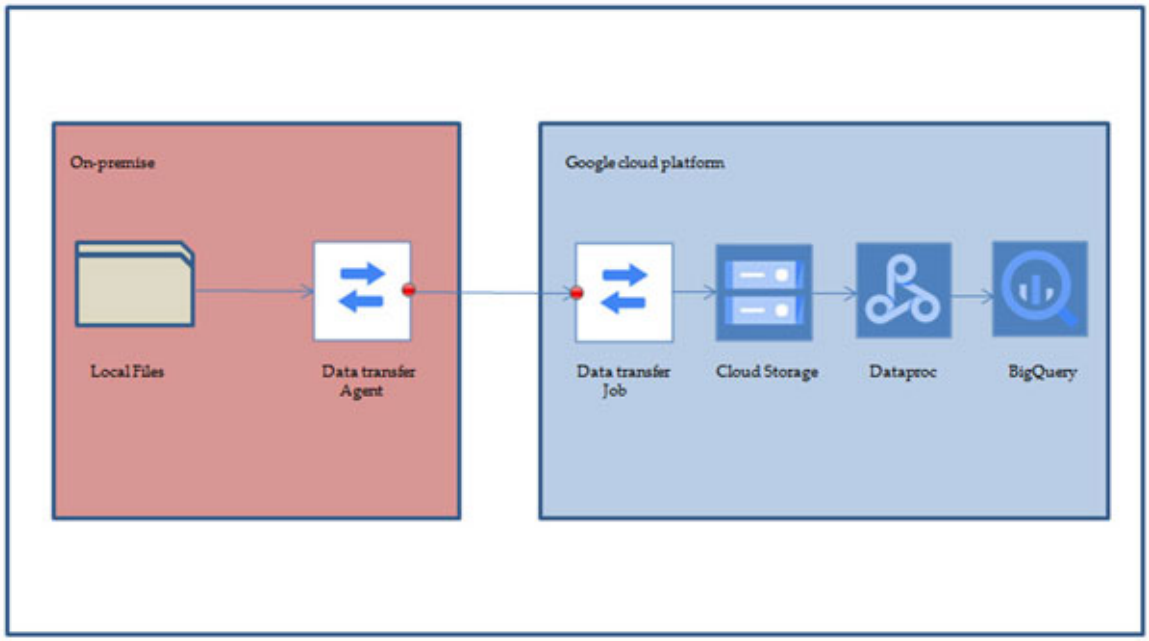

Figure 10.6: *Storage Transfer Service: On-Prem data transfer*

Storage Transfer Service can perform the full one-time transfer or periodic transfer. It also provides you with various configuration options. For example, you get an option to delete the data in the source storage once it has been transferred. It also provides an option to delete an object in the destination bucket in case it is not available in the source anymore. By default, it uses TLS encryption for any HTTPS connection.

Features of Storage Transfer Service

Some of the important features of Storage Transfer Service are as follows:

- **Integrity**: One of the important features of a Storage Transfer Service is that it maintains data integrity while transferring the data. STS uses metadata from the source system to ensure that there are no changes. It also does a checksum calculation. Apart from this, STS also uses TLS encryption for the HTTPS connection.

- **Scheduling**: STS allows you to create incremental schedules to transfer the data from source to destination.
- **Throttling**: You can adjust bandwidth to avoid any latency issues with the production system.
- **Monitoring and Logging**: You can always monitor the progress of data transfer jobs. Transfer logs for any On-Prem data transfer help you in verifying the results of the data transfer.
- **Scale-out option**: You can use multiple storage transfer service agents to improve the performance of data transfer.
- **Fault tolerance**: If some of your agents fail, then others can still continue with the data transfer. Apart from this, when your agent comes back, the transfer job will restart from the point where it got stopped.

Database Migration Service

Database Migration Service is a service offered by Google Cloud Platform that is used to migrate, modernize or simply replicate databases from On-Prem or cloud to GCP. It allows you to do as-is migration from MySQL, SQL server, and PostgreSQL to Cloud SQL. Apart from this, it also helps you in modernizing and migrating your Oracle workload to Cloud SQL for PostgreSQL. Database Migration Service provides you easy access to use **Graphical User Interface(GUI)**, using which you can perform migration in a few clicks. Apart from Google Cloud Console, you can also use Cloud CLI and Database Migration Service API to perform database migration.

Features of Database Migration Service

Some of the important features of database migration services are as follows:

- **Security and Encryption**: Database Migration Service provides default encryption for the data being migrated and also secure connectivity options to perform the migration.
- **Modernization**: Database Migration services supports DB modernization by helping in converting from Oracle database to PostgreSQL. Database Migration Service gets integrated with the Ora2Pg tool, which is an open-source tool for schema conversion.
- **Serverless**: Database Migration Service is serverless, because of which we need not to bother about provisioning and managing the service.

- **Ease of use**: Database Migration Service provides easy to use graphical user interface that can be used to configure the migration jobs. It provides a secure connectivity option and validation before migration gets executed.

Elements in Database Migration Service

The main elements in database migration service are as follows:

- **Connection profiles**: Connection profile is the first element in the Database Migration Service. While creating a connection profile, you require to enter some details, such as the hostname or IP address of the source, along with port ID information and credential that will be used to authenticate the source instance.
- **Conversion workspace**: The conversion workspace is the one that helps you in converting schema and code objects from the source database, as per the formatting requirement of the target database instance. This is primarily used in migrating to a heterogeneous database.
- **Migration job**: Once your connection profile setup is complete, you will create a migration Job. While creating a migration job, the main components that you specify are the source database, target database, connection profile, database version, migration job type, and so on.

Figure 10.7 illustrates the database migration service:

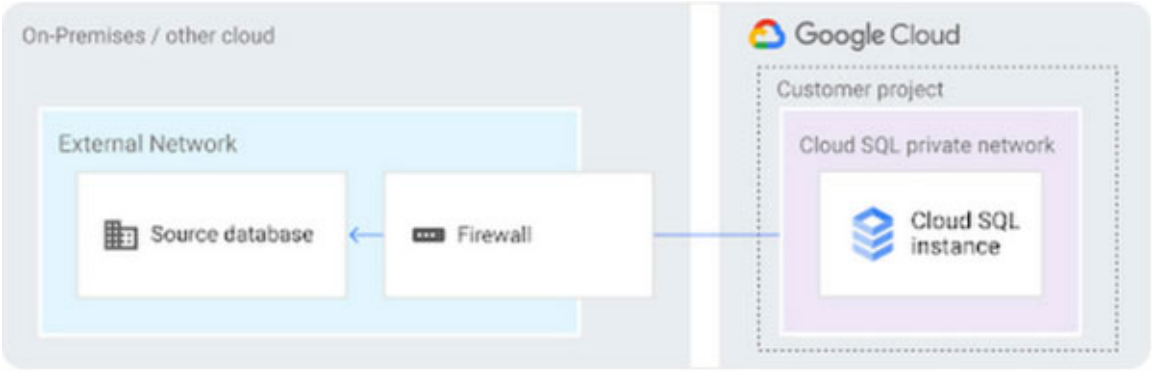

Figure 10.7: *Database Migration Service.*
Source: *https://cloud.google.com/blog/topics/developers-practitioners/database-migration-service-connectivity-technical-introspective*

Use cases

Some of the use cases for database migration services are as follows:

- **Homogenous database migration**: Database Migration Service can be used to perform homogenous migration, that is, from MySQL, MSSQL, and PostgreSQL to CloudSQL for MySQL, CloudSQL for MSSQL, and CloudSQL for PostgreSQL in GCP, respectively.
- **Heterogeneous database migration**: Database Migration Service can be used to migrate and modernize Oracle-based workload to PostgreSQL. In this case, there will be schema and formatting changes required to migrate the database.
- **On-Prem database migration**: You can migrate self-hosted DB in a VM to a managed database service in the Google Cloud Platform.
- **Multi-cloud continuous replication**: You can setup a continuous replication from any third-party cloud service provider to the Google Cloud Platform. Please note that while performing this replication, only the primary copy will be writable, and the other copy will be read-only.

BigQuery Data Transfer Service

BigQuery Data Transfer Service is a managed service that is used to transfer data to BigQuery. It can schedule your data transfer from various sources. This service is an integral part of the BigQuery service itself. You can move data from many **Software As A Service** (**SAAS**) applications, Amazon S3 storage, and data warehouses services like Teradata and Amazon Redshift. Apart from this, BigQuery Data Transfer Service also supports third-party transfers from external data sources such as **Adobe analytics, Facebook ads**, and so on.

The lists of supported sources are as follows:

Google Software as Service applications

- YouTube Channel reports
- YouTube Content Owners reports
- Campaign Manager
- Google Merchant Center (beta)
- Cloud Storage
- Google Play
- Google Ad Manager

- Google Ads
- Search Ads 360 (beta)

External cloud storage providers:

- Amazon S3

Data warehouses:

- Teradata
- Amazon Redshift

Elements in BigQuery Data Transfer Service

The major elements in BigQuery Data Transfer Service are as follows:

- **Creating transfer**: BigQuery Data Transfer Service is a multiregional service; however, you can configure it in a single region. When you create a destination dataset in a specific region. BigQuery Data Transfer Service configuration is also stored in the same region. Creating a destination dataset is an important step for transferring the data.
- **Source data**: There can be a different types of sources. It can be Google SaaS applications, data coming from Amazon S3 object storage, data warehouse, or some third-party sources.
- **Schedule**: Using scheduling, you can specify the date and time during which data should be copied from source to destination. There are increment schedules that you create for copying the data.
- **Schema**: There are two ways of defining the schema in the destination dataset. You can either opt for a default schema, or you can customize a schema for the target dataset. The default schema is extracted from the data source itself. It also depends on the connector that is used to connect to the data source.
- **Destination dataset**: The destination dataset is the dataset that you have already created in BigQuery, and where this data will be moved. There is a dataset ID associated with it when you configure the transfer.

Use cases

A few use cases for BigQuery Data Transfer Service are as follows.

- **From Amazon S3 bucket to BigQuery Transfer**: BigQuery Data Transfer Service supports transferring data from Amazon S3 buckets to BigQuery. However, source data should be in CSV, JSON, Avro, Parquet, or ORC format. For Avro, ORC, and Parquet, the BigQuery Data Transfer Service also supports data in a compressed form. The pre-requisites for transferring data from the Amazon S3 bucket to BigQuery are as follows:
    - Target Dataset in BigQuery
    - Amazon S3 URI
    - Access key ID
    - Secret Access Key
    - AmazonS3ReadOnlyAccess as a minimum policy for the source data in S3
    - Service account in GCP with the required permissions

Figure 10.8 illustrates the Amazon S3 to BigQuery migration:

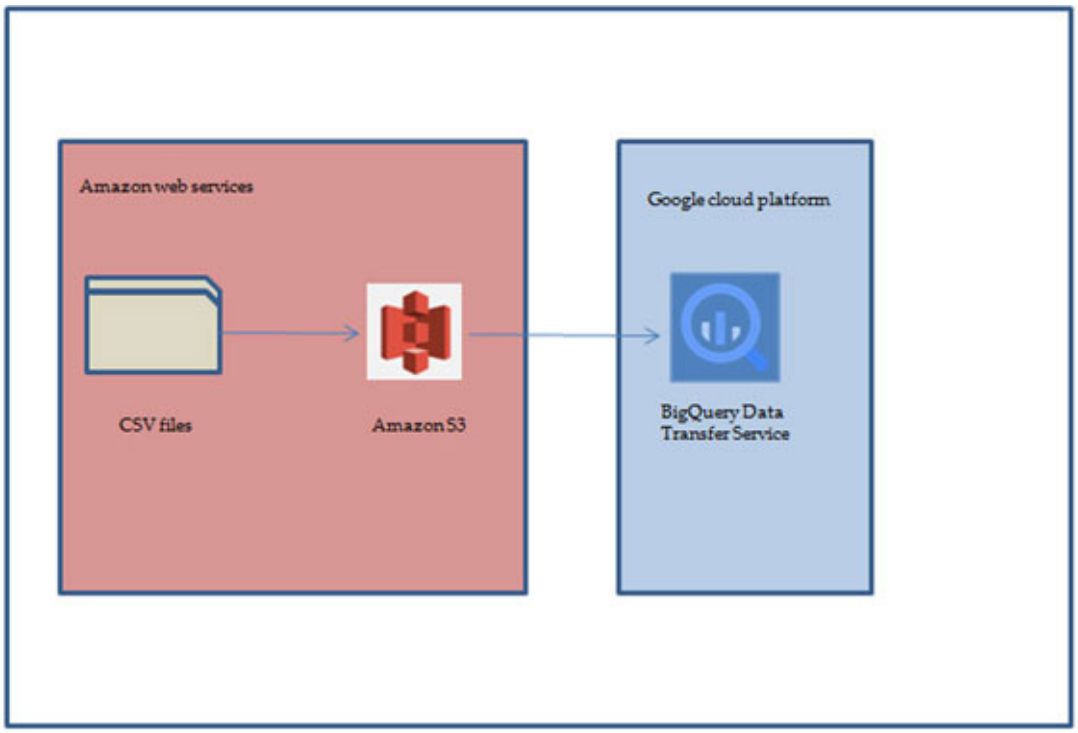

Figure 10.8: *Amazon S3 to BigQuery migration*

- **Teradata to BigQuery transfer**: Migrating data from an external data warehouse service such as Teradata to BigQuery is another interesting use case for BigQuery Transfer Service. We require a special migration agent for migrating from Teradata to BigQuery. Some of the pre-requisites are as follows:
    - Target Dataset in BigQuery
    - Cloud Storage buckets in the GCP cloud for staging the data
    - A local machine with migration agent and JDBC connection with Teradata instance. Local machines will also require Teradata tools such as BTEQ and FastLoad
    - Custom Schema for schema conversion
    - Service account in GCP with the required permissions

Figure 10.9 illustrates the Teradata to BigQuery migration.

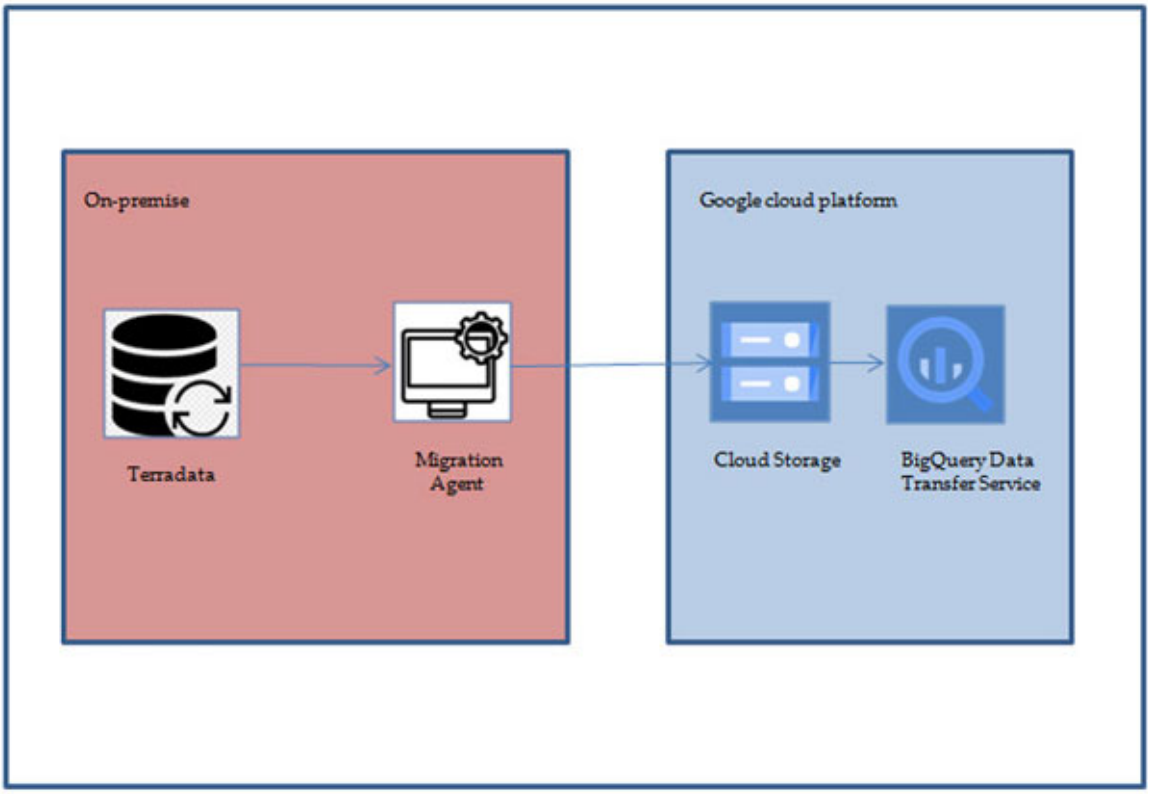

Figure 10.9: *Teradata to BigQuery migration*

Lab exercises

Let us now work on a few lab exercises.

Migrating MySQL database instance to cloud SQL using database migration service

In this lab exercise, we will practice how to migrate a MySQL database instance to Cloud SQL using a database migration service. In our lab, we will have a MySQL database instance already residing on a VM in the Google Cloud Platform. Learners are requested to have the MySQL database installed in the source VM. In a more realistic scenario, you will have an On-Prem MySQL instance that needs to get migrated to Cloud SQL. Follow the following steps:

1. Login to your Google Cloud Console and check if Database Migration API is enabled by entering Database Migration API in the search bar**.** If not, then enable it. Refer to *figure 10.10*:

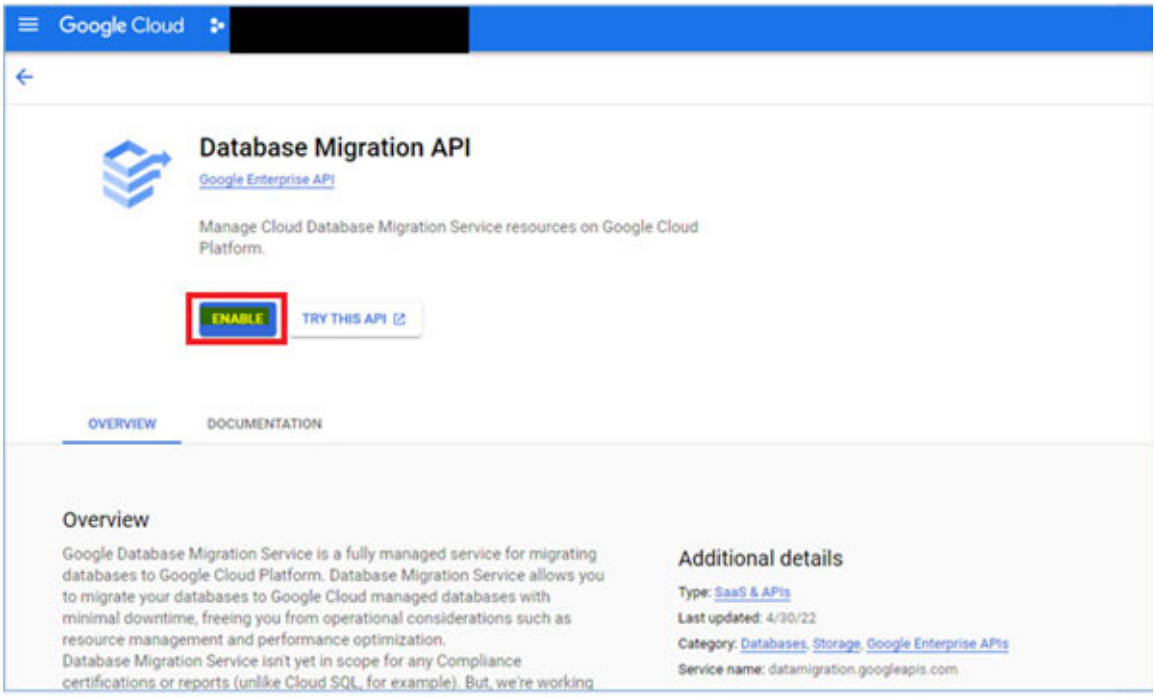

Figure 10.10: *Enabling database migration API*

2. Now, we will try to get the IP address from the source instance. Go to **navigation menu** |`Compute Engine`, and select `VM instances`, as shown in *figure 10.11*:

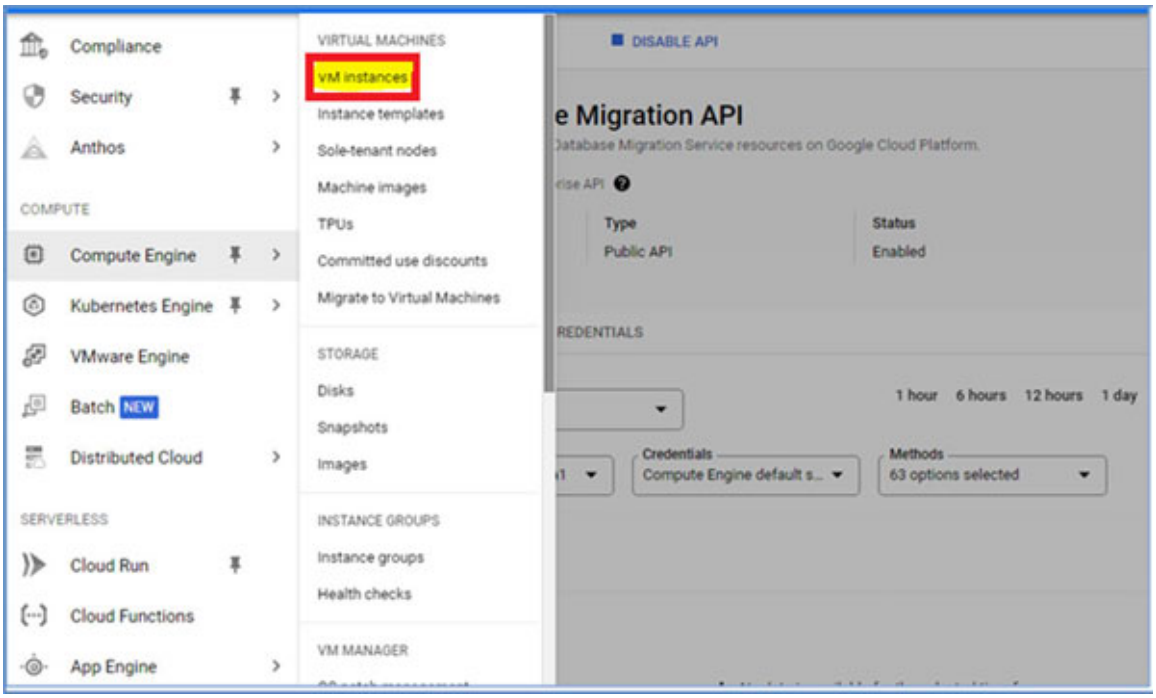

Figure 10.11: *Open VM instances*

3. Copy the **Internal IP** of the VM instance, as shown in *figure 10.12*:

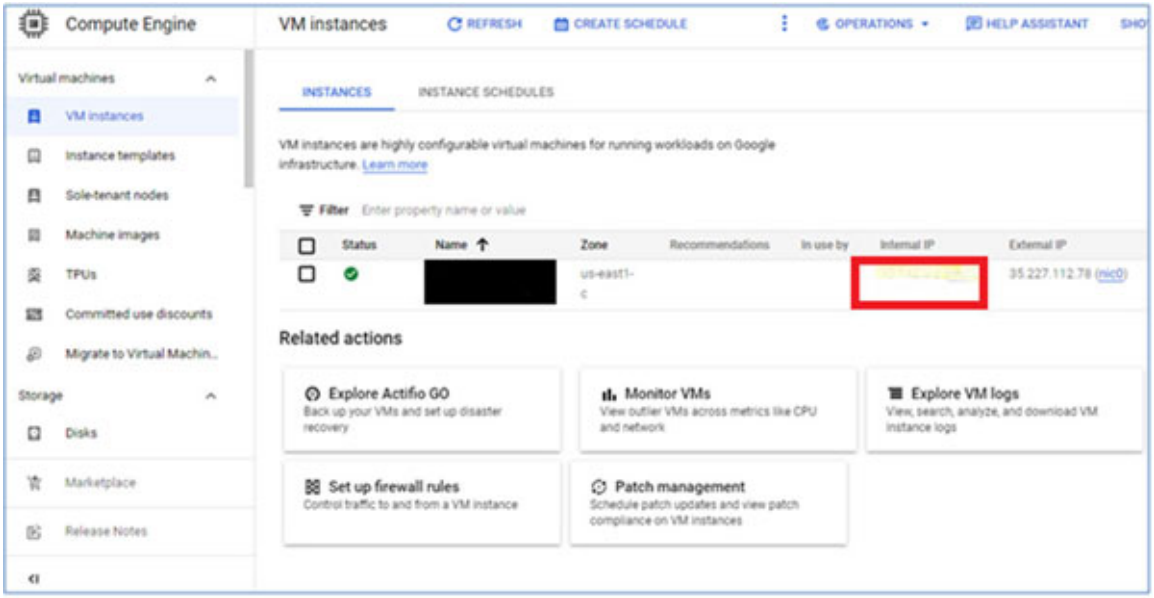

Figure 10.12: *Copying the internal IP of VM instance*

4. Now, we will create the connection profile for the source database instance. Go to **Navigation Menu** |`Database migration` |`Connection profiles`, as shown in *figure 10.13*:

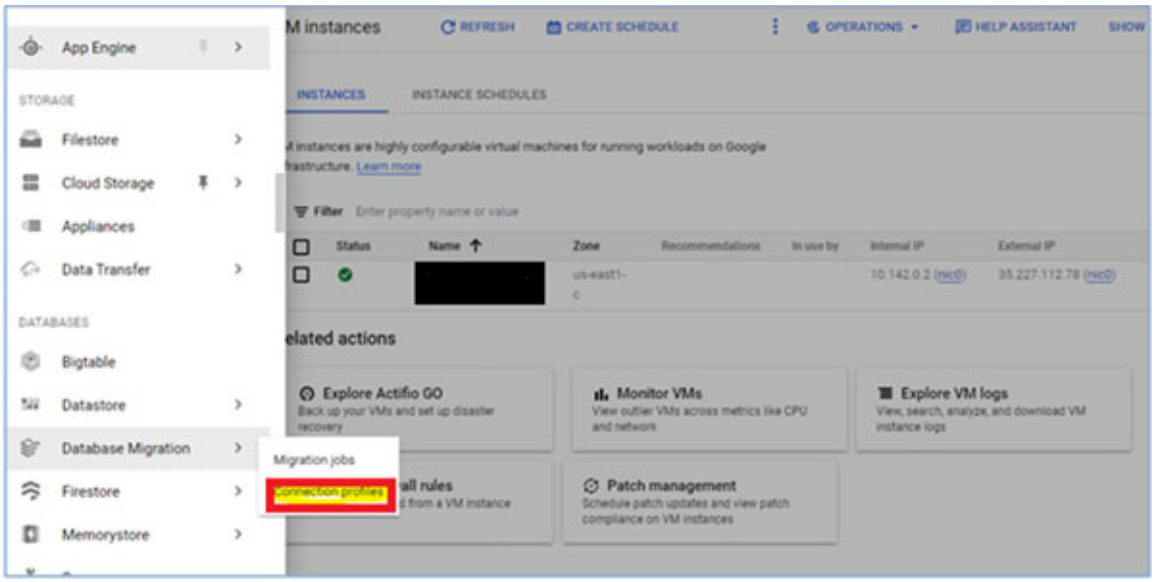

Figure 10.13: *Open connection profiles*

5. Click on **`CREATE PROFILE`**, as shown in *figure 10.14*:

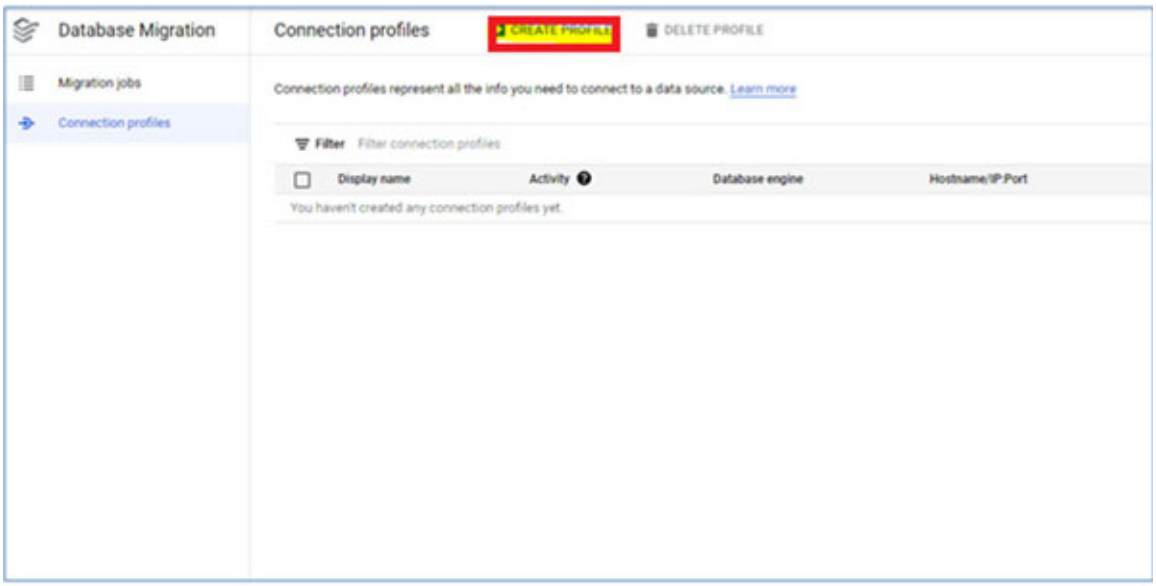

Figure 10.14: *Creating a profile*

6. Select MySQL as the Database Engine. Give the connection profile a name and leave the connection profile ID to a default value. Enter the internal IP of the source database that you have copied from the previous step for the Host Name or IP address field. Enter username and password. This is the username and password for the source **database**. Enter region as **`us-east1`**.Select **the** **`Encryption type`** as **`None`**. Click on **`Create`**. Refer to *figure 10.15*:

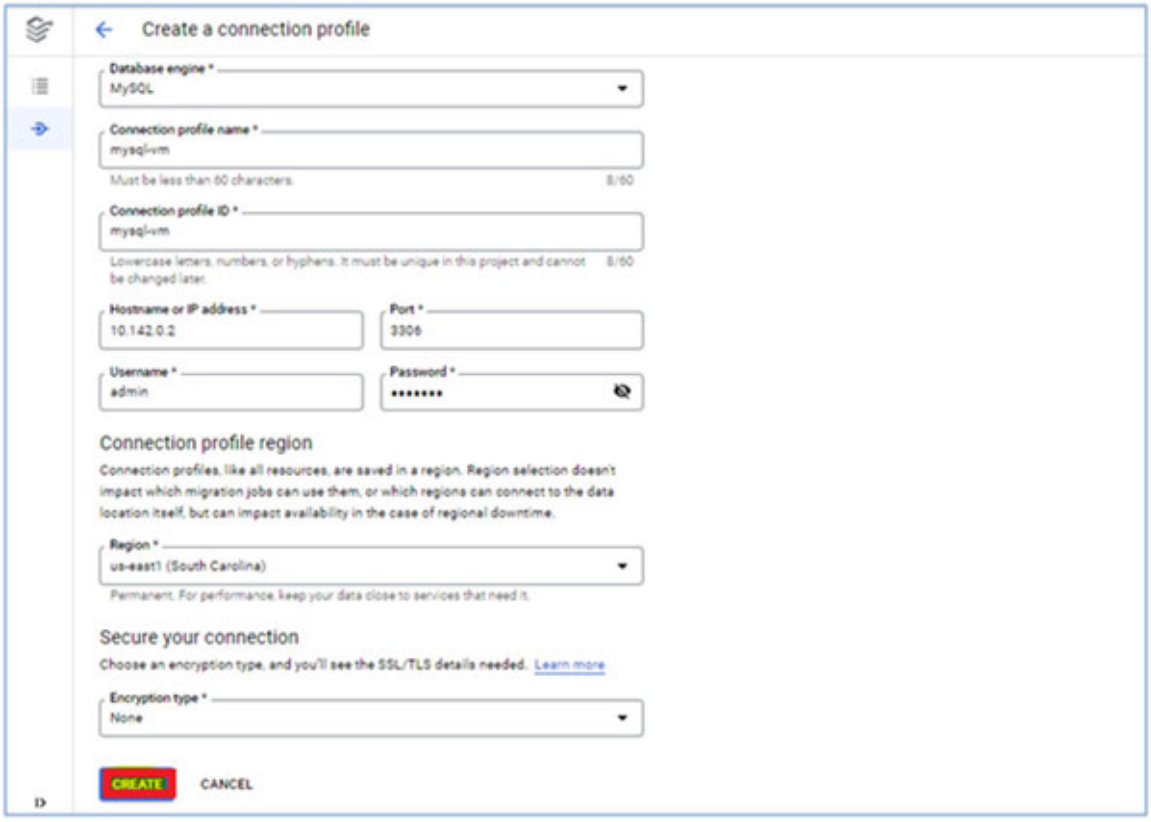

Figure 10.15: *Fill in details to create a connection profile*

7. Now, you should be able to see your connection profile listed under **`Connection profiles`**, as shown in *figure 10.16*:

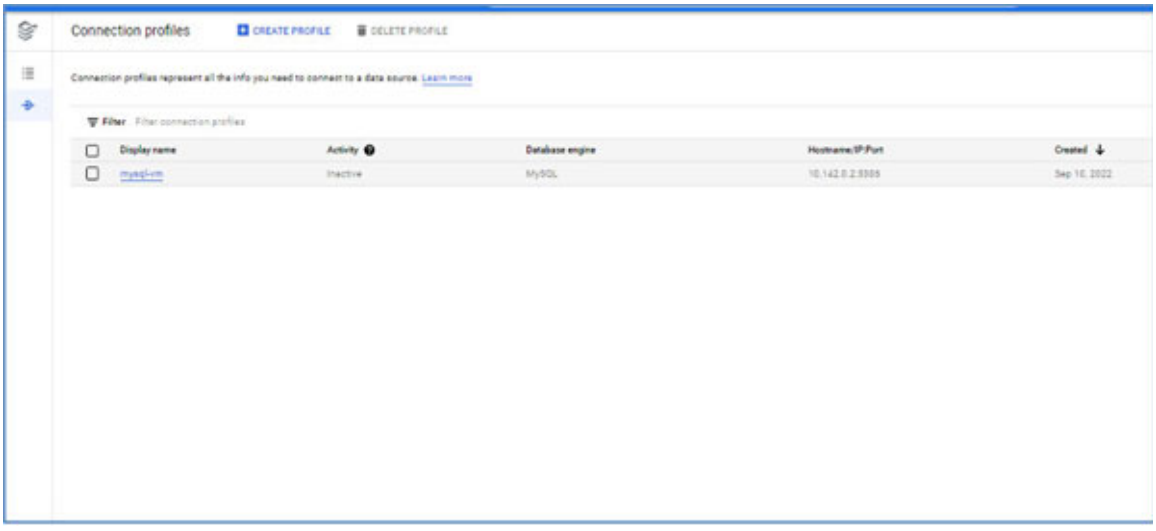

Figure 10.16: *Connection profile is listed*

8. Now, we will create a migration job to perform continuous migration from the source database to the cloud SQL target. In Navigation Menu,

select **`Database Migration`** |**`Migration Jobs`**. Refer to *figure10.17*:

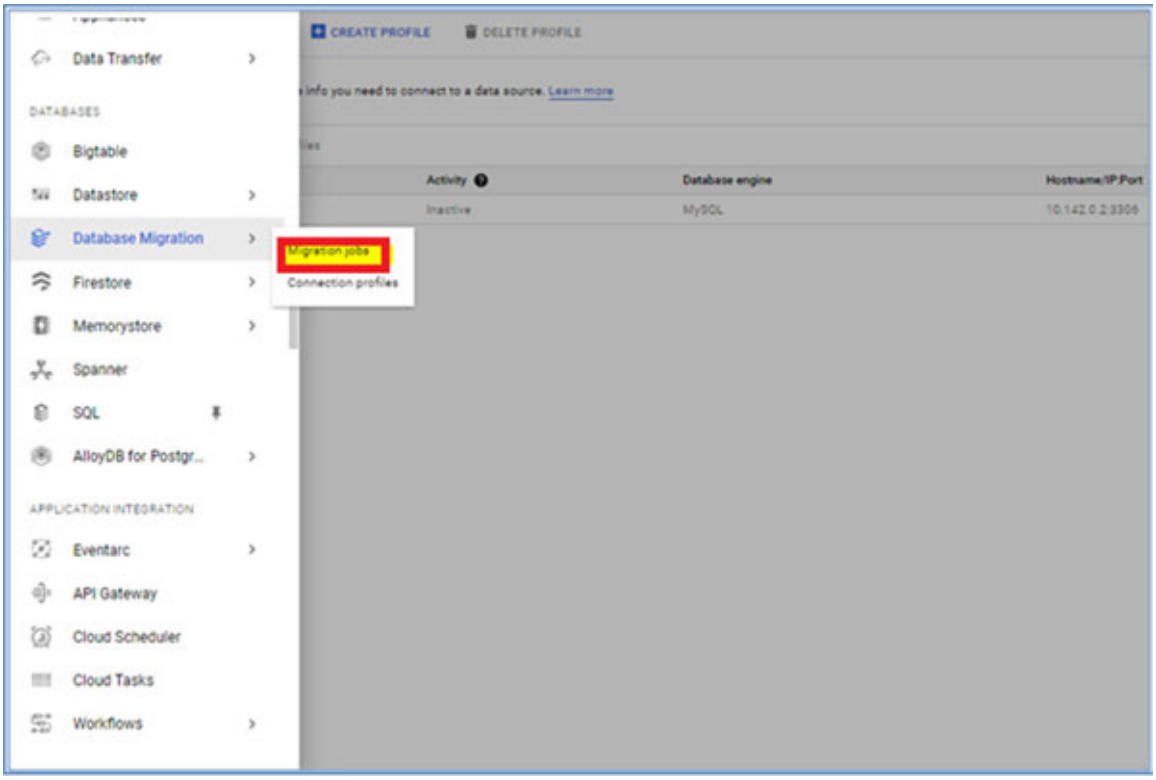

Figure 10.17: *Open migration Jobs option*

9. Now, click on **Create Migration Job**. Enter a name for Migration Job, Select **`MySQL`** as Source database Engine, and Select **`Continuous`** under **`Migration Job Type`**. Click on **`Save and Continue`**. Refer to *figure 10.18*:

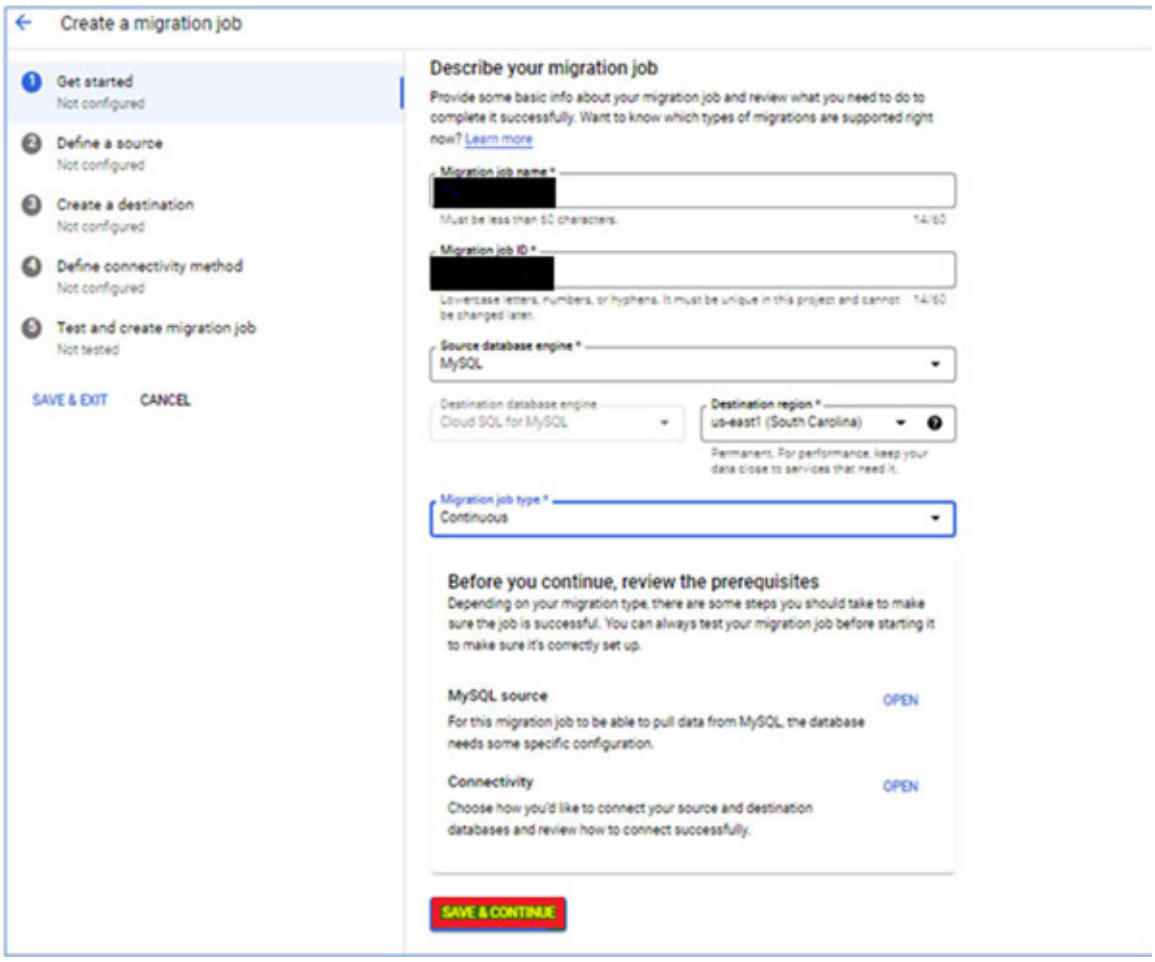

Figure 10.18: *Fill in details to create a migration job*

10. On the next page, select **`the source connection profile`** that we have already created, and click on **`Save and Continue`**, as shown in *figure 10.19*:

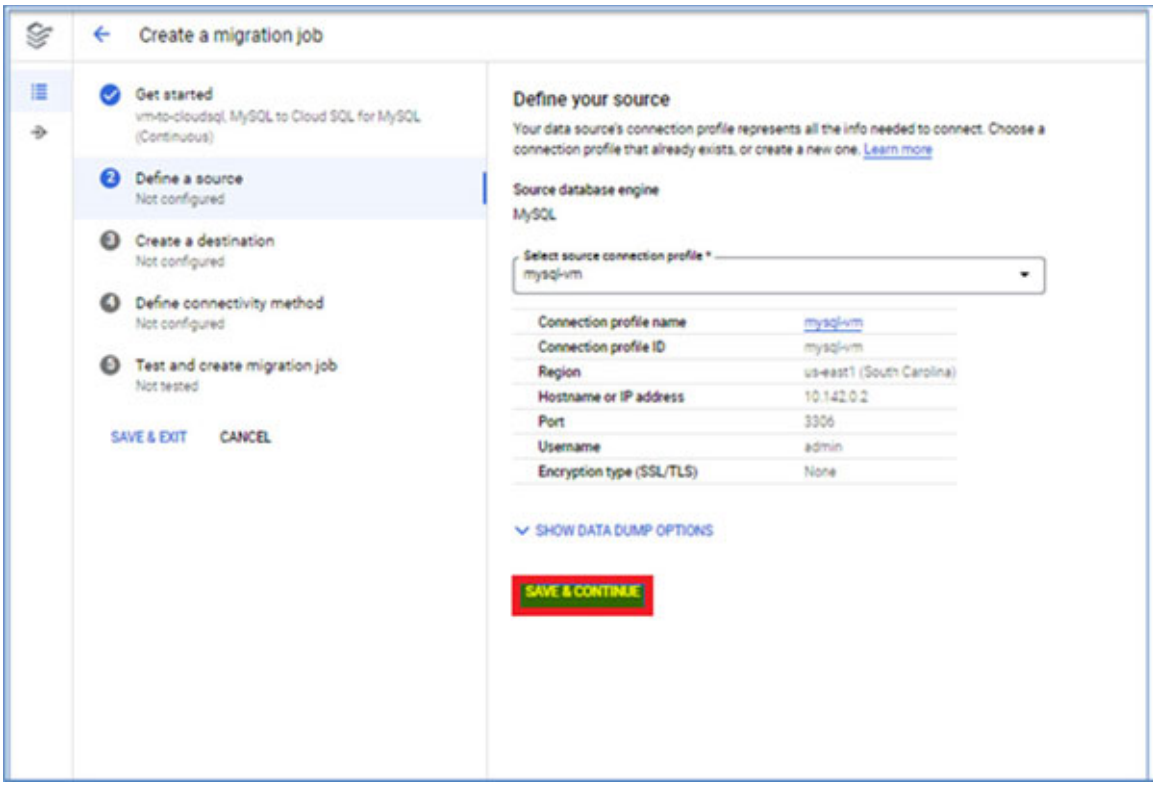

Figure 10.19: *Selecting source connection profile*

11. Now, on the next page, we will create a CloudSQL destination instance. Insert some names under the **Destination instance ID.** Select some **password** of your choice, or you can opt for **Auto Generated** password**.** Under Zone, please select **`us-east1-c`.** Refer to *figure 10.20*:

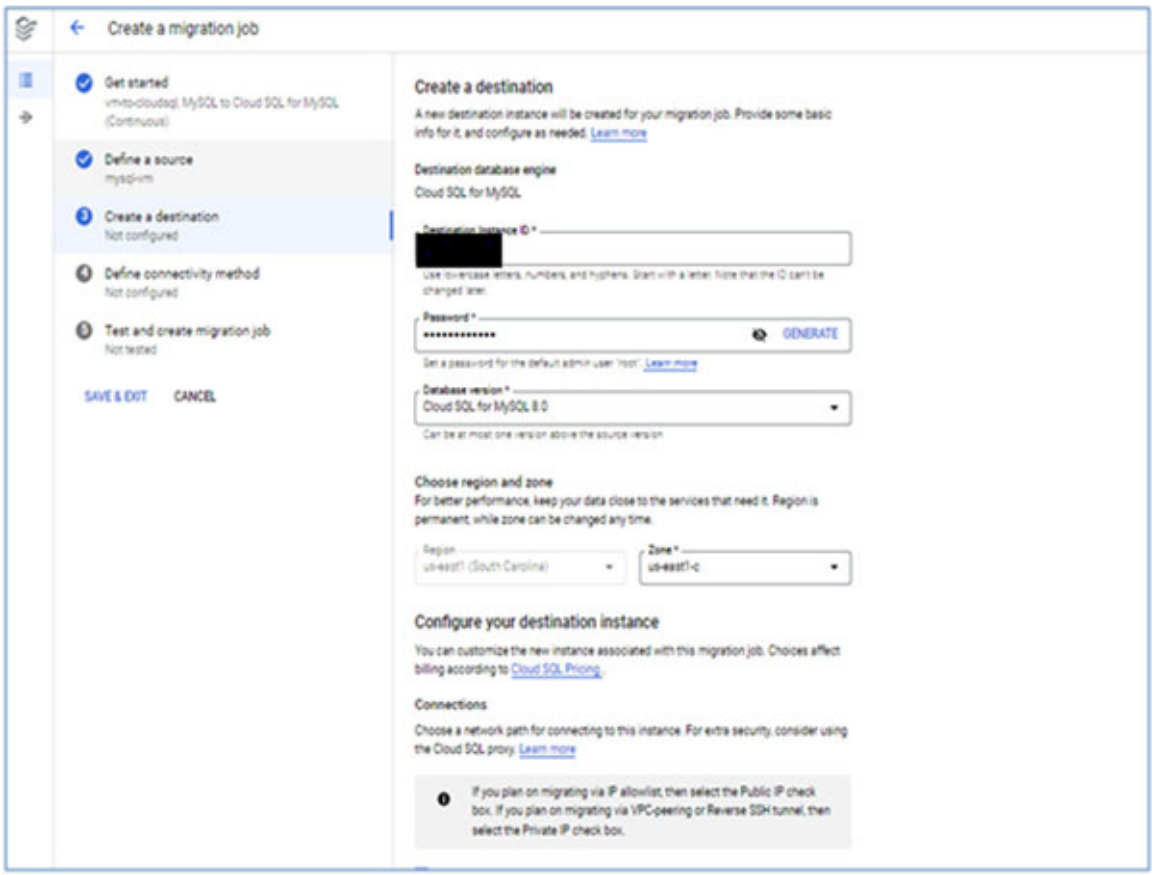

Figure 10.20: *Creating a CloudSQL destination instance*

12. Checkthe **`Private IP`** and **`Public IP`** checkboxes. Select **`Use an automatically allocated IP range`**, and click on **`Allocate &connect`**. It will take some time for this step to get complete. Refer to *figure 10.21*:

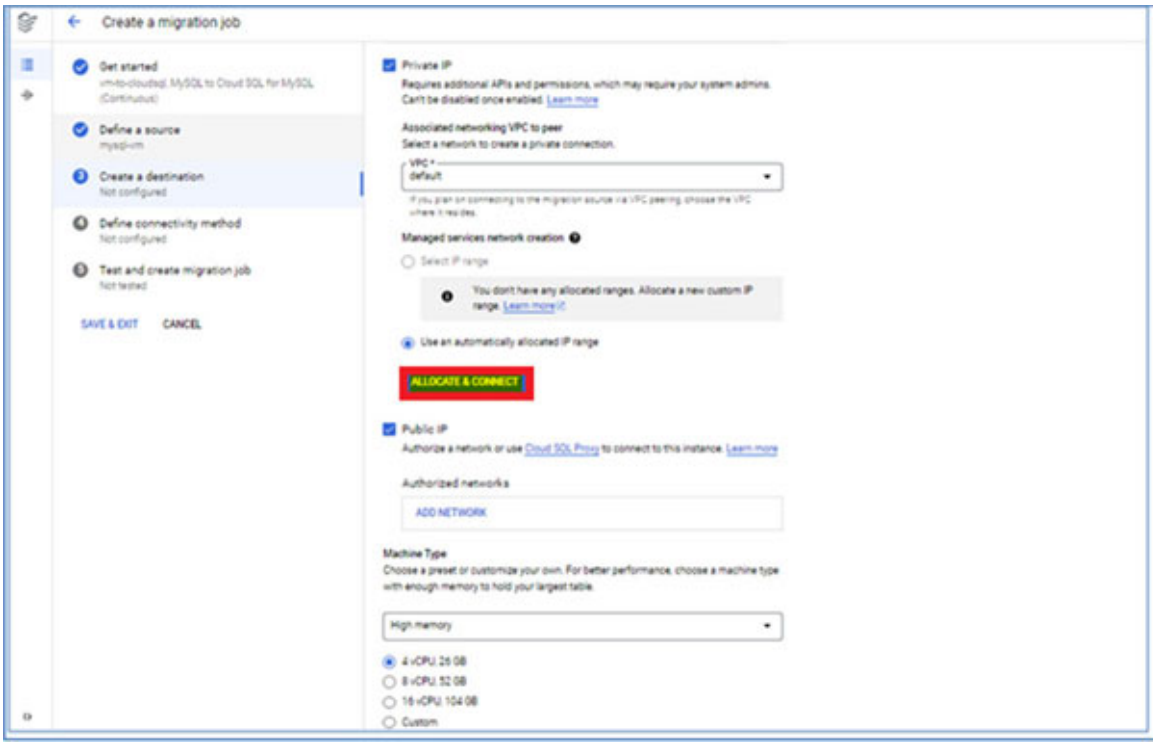

Figure 10.21: *Fill in details to create a destination*

13. Leave other values as default and click on **`Create & continue`**. This step will take some time to complete. Refer to *figure 10.22*:

Figure 10.22: *Destination will be created*

14. Now, we will define the connectivity method. Select **VPC peering** under **Connectivity Method** and **default** under **VPC**. Select **Configure & Continue**. Refer to *figure 10.23*:

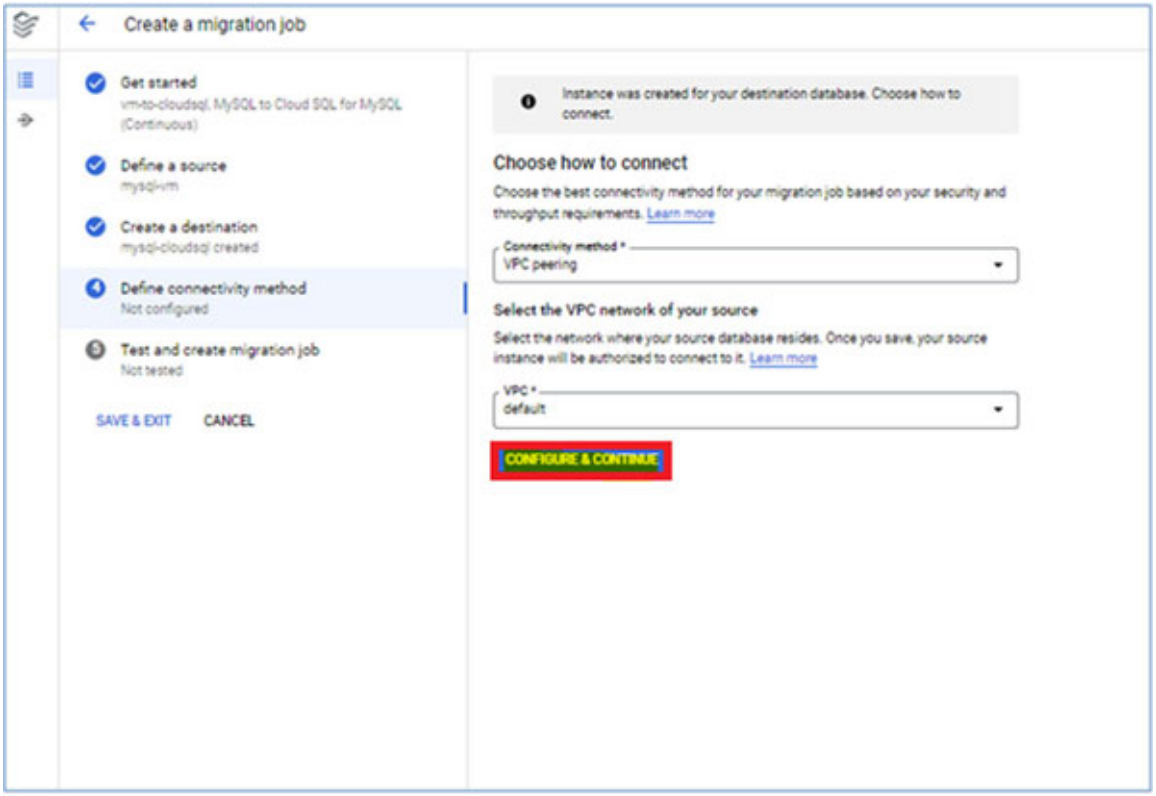

Figure 10.23: *Defining connectivity method*

15. Now, we will test and start the migration Job. Click on **`Test Job`**. It will test the migration job that has been created. Once the job has been tested successfully, click on **`Create and Start Job`**, as shown in *figure 10.24*:

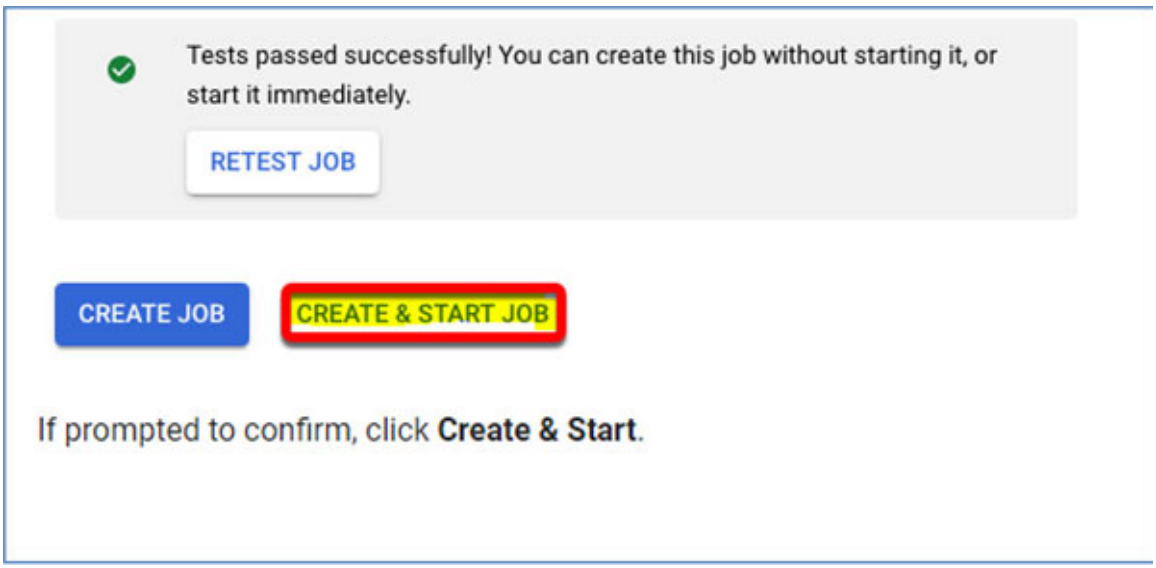

Figure 10.24: *Creating and starting the job*

16. Once you go to **Navigation menu** |`Database migration` |`Migration Jobs`, you should be able to see your migration job running, as shown in *figure 10.25*:

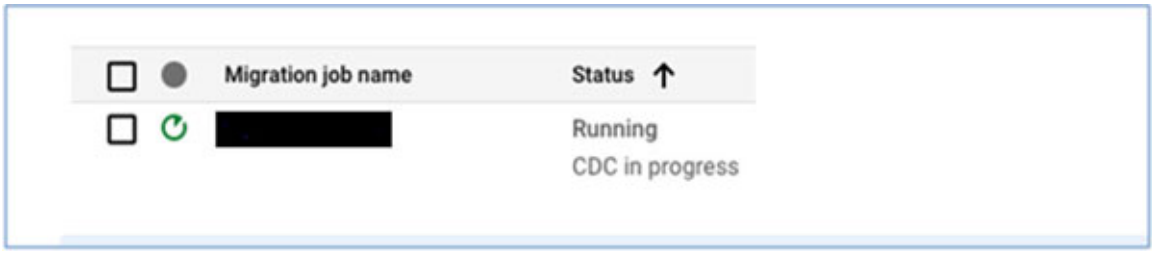

Figure 10.25: *Migration job is running*

Conclusion

In this chapter, we have discussed about various migration options available in the Google Cloud Platform. There are many customers who are looking to move their On-prem workload to the cloud. Due to this, there is a huge demand for skilled professionals in this area. Readers are advised to go through the GCP site and other online study material to develop a further understanding of this.

Key terms

- **POC**: Proof of concept. It is an exercise to validate and test any activity that we are going to perform using some samples.
- **Schema**: In the context of a database, a schema is a structure of a database that defines tables, fields, data types, and relationships between different entities.
- **TLS**: TLS stands for Transport Layer Security. It is an encryption protocol used in internet communication.

Questions

1. What is migration? What are the various phases of migration activity?
2. What do you understand by POC, and why it is done?
3. What are various database migration tools available in the Google Cloud Platform?

Further readings

- **https://cloud.google.com/bigquery/docs/dts-introduction**
- **https://cloud.google.com/storage-transfer-service**
- **https://cloud.google.com/database-migration**
- **https://cloud.google.com/migrate/virtual-machines**
- **https://cloud.google.com/transfer-appliance/docs/4.0/overview**
- **https://cloud.google.com/migrate/stratozone/docs/about-stratozone**

CHAPTER 11

Best Practices

Introduction

Best practices can be defined as a set of guidelines that we need to follow in order to perform a task and achieve the desired results. Best practices are developed over a period of time as a result of various user experiences. These are procedures that are considered the most effective in order to achieve the desired results.

Structure

In this chapter, we will cover the following topics:

- Landing Zone—best practices
- Infrastructure provisioning—best practices
- Cloud IAM—best practices
- Migration—best practices

Objectives

The objective of this chapter is to familiarize readers with some of the best practices specific to GCP services. We have covered some of them in the previous chapters, and here, we are going to discuss a few more. These best practices will help you in making the best decisions while designing or provisioning resources in the Google Cloud Platform.

Landing Zone—best practices

The landing zone is a foundation configuration for your cloud adoption environment. **Google Cloud Platform** (**GCP**) helps you with deploying, securing, and scaling Google Cloud services. You can deploy it once you

have defined organizational resources and billing accounts for your target GCP environment.

Landing zone helps with creating a standardized way of provisioning cloud resources. In a typical Google Cloud setup, we have the following components that are part of the Landing zone:

- Resource hierarchy with Organizational policies.
- Cloud identity account, IAM roles, and policies.
- Shared VPC that has network, security configuration, and shared services configured in the host project. This should be implemented in all the target environments (Prod, Dev, Test, DR, and so on).
- Service projects deployed in a Shared VPC that have resources configured in it.
- Cloud VPN.
- Cloud NAT.
- Any connectivity requirement with an On-Prem data center or any third-party cloud provider.
- Cloud DNS.
- Cloud Monitoring and Logging.

Figure 11.1 shows a typical landing zone example:

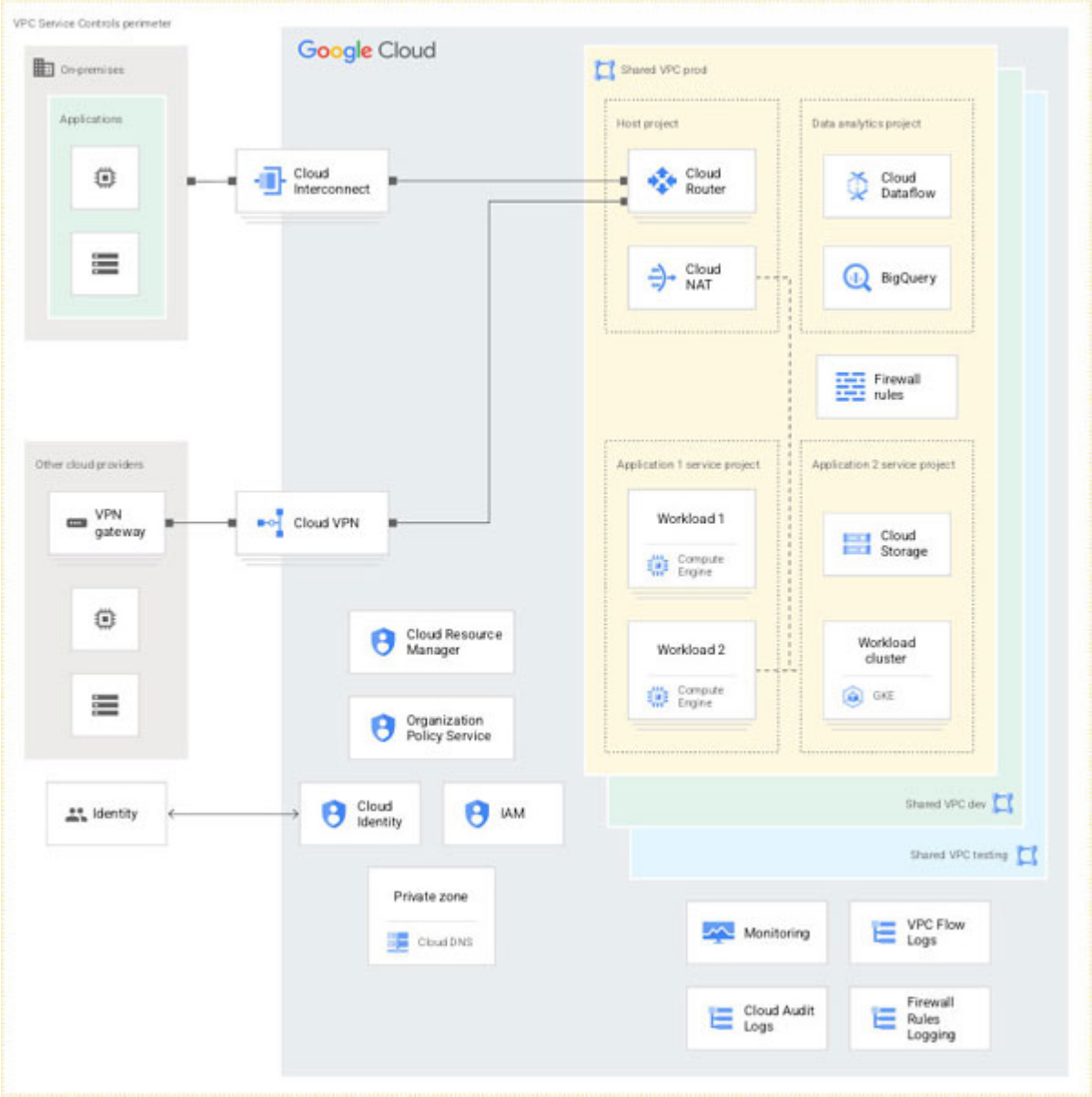

Figure 11.1: *Typical landing zone example.*
Source: *https://cloud.google.com/architecture/landing-zones*

Best practices

- **Understand the business requirements**: It is very important to understand business requirements before designing a landing zone. Business criticality, compliance, and security needs should be properly understood, and accordingly, any security, networking, and environmental decisions should be made to design the landing zone.

- **Implement Initial design**: It is advised to implement the initial design for the primary workload and then evolve as we proceed further.
- **Use Infrastructure as a code**: It is recommended to use Infrastructure-as-a-Code for implementing the landing zone. For example, Terraform can help you in designing a reusable and modular code to implement a Landing zone. A **Continuous integration/Continuous deployment** (**CI/CD**) pipeline that uses a repository can be used to store the code and deploy the landing zone.
- **Right team with the right resources**: We need to make sure that all the required stakeholders are involved from the designing phase itself so that all aspects of landing zone design and implementation are taken care. We would require a subject matter expert for network, security, and identity management, a project manager, and people representing the project and business owner. It will help us with implementing correct environments, security policies, access policies, and so on.

Infrastructure provisioning—best practices

Following are some of the best practices while doing infrastructure provisioning:

- **Use Spot VMs whenever required**: It is recommended to use Spot VMs if the workload that you are running is supposed to run for a small amount of time. This results in significant cost savings. Spot VMs are ideal when your workloads can handle faults and sudden instance termination. One of the example of such a workload is running batch processing jobs.
- **Use automation for infrastructure provisioning**: It is recommended to use automation tools such as Terraform to do infrastructure provisioning. Terraform templates can be used with some slight modifications for any other similar requirements. Moreover, it helps with a significant reduction in manual efforts required for provisioning the resources.
- **Use rightsizing recommendations**: Compute Engine gives rightsizing recommendations on the basis of metrics generated by the monitoring

tools. It is a good practice to optimize resources as per the recommendations.

- **Disable auto delete for the persistent disk**: It is recommended to disable auto-delete option for the persistent disk attached to your VM, to avoid data loss for production and mission-critical applications.
- **Cross-verify publicly accessible Cloud Storage**: Cross-check your Cloud Storage buckets for any public accessibility.
- **Select the correct storage class:** Make sure that you are using the correct storage class for your object storage requirement. It helps with cost optimization.
- **Monitor persistent disk snapshot**: We create persistent disk snapshots for performing the backup. Please ensure to delete them if they are no longer needed. You can set up a policy defining how many snapshots need to be retained for a persistent disk.
- **Make use of discount options for resources**: Compute Engine comes with committed and sustained use discounts. Make sure that you make use of these discounts, considering your requirement.
- **Opt for the correct DR Strategy**: Opt for the appropriate **Disaster Recovery(DR)** strategy that fits your **Recovery Point Objective/Recovery Time Objective(RPO/RTO)** requirement. A Hot DR is the one in which your DR setup is always ready, and you just need to perform a failover. However, the cost of managing a DR setup is significant in this case. It will be used when your RPO/RTO requirements are stringent. A warm DR is one where your resources in the DR are partially provisioned. A cold DR is where most of your resources are not provisioned during normal times, However, you can bring them up using an automated script. This DR is used when your RPO/RTO requirements are not that stringent. As resources are not consumed all the time, this results in significant cost savings.
- **Enable monitoring for the resources**: It is a common practice to monitor resources. It helps us with identifying any issues with the resources that are being consumed.
- **Server autoscaling**: Make use of server autoscaling by using managed instance group with a Load balancer. We should do this for eligible or necessary workloads.

- **Enable secure boot**: It is recommended to enable secure boot if the third-party unsigned kernel modules are not used.
- **Shielded VM**: Shielded VMs secure your VMs against rootkits and bootkits. Rootkits are programs that give unauthorized access to your VMs, and bootkits are an advanced form of rootkits. It is suggested to use shielded VMs to protect against these attacks.

Cloud IAM—best practices

Following are some of the important cloud IAM best practices:

- **Do not use primitive or basic roles**: It is suggested not to assign primitive roles as they are too permissive and give unrequited permissions to the users.
- **Principle of least privilege**: Any applications, users, or groups, should only be assigned roles that are required to perform a task.
- **Do not use a personal Gmail account for access**: Always use corporate login credentials and not a personal Gmail account to login.
- **No administrative access to service accounts**: It is recommended not to give administrative access to your service accounts.
- **Rotate IAM service account keys**: It is recommended to rotate IAM service account keys. It helps to avoid security breaches in case if keys are compromised.
- **No external users to be added to an existing Group**: It is recommended not to add any external users to a Group. It is required to avoid accidentally acquired permissions.
- **Use groups wherever possible**: Use groups to manage permissions for internal users wherever possible, as it will avoid unnecessary management overhead of managing users individually.
- **Auditing**: Use and export audit logs for tracking changes in your IAM policy.
- **Custom role**: If there are no predefined roles that fulfill your requirement, use a custom role instead.
- **Folder-level access**: If a user or group needs access across multiple projects, it is advised to set the role at the folder level.

DevOps and automation –best practices

In *Chapter 6, Security and Monitoring Services in Google Cloud*, we have already discussed about the concepts of DevOps. Here we will discuss about some of the best practices that need to be followed while leveraging DevOps and automation.

Following are some of the best practices:

- **Implement CI/CD pipeline**: It is recommended to use a continuous integration/continuous delivery pipeline for code integration and deployment. With the help of CI, developers can merge the code frequently to the repository and use automated tools to test the correctness of the code. Continuous delivery helps with deploying the application code to the environments without hassle.
- **Monitoring**: It is important to monitor the DevOps pipelines for any build or deployment issues. Any issue with the DevOps pipeline will result in unnecessary delays in production deployment.
- **Integration with the existing open-source tools**: It is a good practice to use a combination of open-source tools, such as Terraform, Jenkins, and cloud-native tools, like container or artifact registry, to reap the benefits of both worlds. Using open-source tools help with cost reduction and cloud vendor lock-in.
- **Use a step-by-step approach**: It is suggested to implement DevOps and automation one step at a time. It helps with building the confidence of relevant stakeholders and avoids any major issues affecting the whole environment.
- **Have a dedicated team with skilled resources**: It is suggested to have a dedicated DevOps team in order to work on DevOps and automation requirements. This team should have skilled resources with experience in various aspects of DevOps practices.
- **Security controls and tools**: It is recommended to implement proper security controls and integrate security tools in CI/CD pipeline. Proper version control should also be implemented on templates or blueprints.

Migration—best practices

We have already discussed in detail about the migration concepts in *Chapter 10, Migration Services in GCP*. Here, we will touch upon some of the best practices that need to be followed during the migration assessment and execution phases:

- **Involve all required stakeholders**: Ensure to include all stakeholders during the assessment phase. This should include application and infrastructure SMEs from the customer's end, architects, and actual migration SMEs who will be involved in migration activities.
- **Updated inventory**: Your inventory should include all the latest details of the workload that is being migrated, that is, application dependency mapping, application to server mapping, platform, network, and security-related details. Having a stale inventory can result in the failure of the migration plan. If the source data center has too many applications and servers, then there is a possibility that the applications or servers are getting added and removed frequently.
- **Application downtime consideration**: It is essential to assess the downtime that applications can tolerate during migration. Your migration plan should be developed, considering that in mind.
- **Rollback strategy**: A proper rollback strategy should be planned and documented. In case there is a migration failure, there should be a plan in place to rollback application availability status to the original state to minimize downtime.
- **Proof of concept:** It is suggested to plan a POC before performing a migration to the actual production environment. You can select some simple applications with less complexity and integration requirement to execute this POC. It will build confidence in the customer and all relevant stakeholders for a successful migration and will also help in identifying any issue with the migration before the actual migration.
- **Update support team about the migration schedule**: It is suggested to alert the support team about the migration schedule so that we have all the support staff in place if something fails during the actual migration.
- **Monitoring and alert**: Please ensure that your monitoring, logging, and alerting system is in place at the source and the target environment, and all the workloads that are being migrated or have

migrated already should be monitored for any issue. Google Cloud Operations Suite can be used for monitoring and alerting needs in Google Cloud, whereas existing On-prem monitoring and alerting solution should be used to monitor any issues with the workload that still exists in the source environment.

- **Set up a maintenance window to perform the migration**: It is a good practice to set up a maintenance window for performing the migration. Moreover, it is advised to perform migration during off business hours so that there are minimum chances of users getting impacted if something happens during migration.
- **Clean-up of source environment**: It is a good practice to clean up the source environment after a successful migration.

Conclusion

In this chapter, we have discussed a few best practices around Landing Zone, Infrastructure provisioning, DevOps, and Migration. Considering the scope of this book, we have discussed these topics at a high level. As the readers explore more individual topics over a period of time, they will have a better understanding of best practices that need to be followed across different areas. In the upcoming chapter, we will try to explore Bare Metal services available in the Google Cloud Platform.

Questions

1. What is a landing zone? State some of the best practices that you need to follow while designing a landing zone in GCP.
2. State some best practices that you need to follow while doing infrastructure provisioning in GCP.
3. Explain the DevOps concept. Also, mention some of the best practices that need to be followed while implementing DevOps.

Further readings

- **https://cloud.google.com/architecture/migration-to-google-cloud-best-practices**

- **https://cloud.google.com/architecture/migrating-vms-migrate-for-compute-engine-best-practices**
- **https://cloud.google.com/blog/products/identity-security/iam-best-practice-guides-available-now**
- **https://cloud.google.com/architecture/best-practices-continuous-integration-delivery-kubernetes**
- **https://cloud.google.com/recommender/docs/tutorial-iac**
- **https://cloud.google.com/architecture/landing-zones**

CHAPTER 12

Bonus Chapter

Introduction

This chapter is dedicated to some of the offerings that are not included in the previous chapters. These are the services that were either added to the **Google Cloud Platform** recently or could not get placed under some specific topics of chapters that were covered earlier.

Structure

In this chapter, we will cover the following topics:

- Google Cloud VMware Engine
- Oracle—Bare Metal Solution
- Anthos
- Google Distributed Cloud Edge
- Google Cloud Backup and DR

Objectives

The objective of this chapter is to familiarize readers with some of the additional services that were not covered in the previous chapters. Anyone who wants to develop a reasonable understanding of GCP services should have a fair understanding of these recently added services as they make an important part of Google's cloud services portfolio.

Google Cloud VMware Engine

Google Cloud VMware Engine is an offering from Google that lets you run your VMware environment on Google Cloud. It is a fully managed service offered by that leverages VMware technologies such as vSphere, vSAN, NSX, and HCX. Using Google Cloud VMware Engine, you can take

benefit of the cloud consumption model with features like **On-demand consumption** and **pay-as-you-go**. You can still use your existing VMware skills and tools. Using the Google Cloud VMware engine, you can also take benefit of Google's underlying backbone and network that provides you 99.99% availability. VMware Engine runs on a bare metal server infrastructure that can easily integrate with other GCP native services. It runs on a dedicated private cloud that runs on **Hyper-Converged Infrastructure**(**HCI**). Hyper-Converged Infrastructure can be defined as a software-defined infrastructure that combines all elements of a traditional data center, that is, server, networking, and storage. It tightly integrates compute, storage, and network virtualization together and helps with easy management, flexibility, and deployment. With the Google Cloud VMware Engine, there is no need to retrain your technical staff when you move your workload to the cloud, as your team is still using the same components and services that they were using earlier. In Google Cloud VMware Engine, we create a private cloud that comes with a minimum of three nodes and can be extended to 16 nodes in a cluster. You can have maximum of 64 hosts in a private cloud, and there is no limit on the number of private clouds that you can have.

The following are the main components that are part of the VMware Engine:

- **ESXi nodes**: ESXi nodes are the physical nodes that run hypervisor, which provides virtualization capabilities. You can create multiple virtual machines on top of it. These virtual machines will have their own guest operating systems where you run your applications.
- **vCenter server**: VMware vCenter Server is a centralized management software that is used to manage one or more ESXi servers, which can be part of one or more vSphere clusters.
- **VMware tools**: It is a collection of multiple services or tools that help with better management of the VMware environment and easy interaction with the guest operating system.
- **Compute, storage, and memory**: Following are the hardware configuration of the physical servers in which the host operating system is installed. This is the only node type option available in Google Cloud VMware Engine:

    - **Compute**: Intel Xeon Gold 6240 (Cascade Lake), 2.6 GHz (x2), 36 Cores, 72 Hyper-Thread
    - **Memory and Cache**: 768 GB Memory, 2 × 1.6 TB (3.2 TB) Cache
    - **Storage/Data**: 6 × 3.2 TB (19.2 TB) NVMe
- **vSAN**: It is a software-defined enterprise-class storage solution that works on Hyper-converged infrastructure. It is a part of the hypervisor.
- **NSX**: It is a network virtualization and security software created by VMware. It helps in creating a virtual network over a physical network.
- **HCX**: Application mobility platform used to simplify application migration, load balancing, and business continuity across data center and cloud.

Note: A detailed explanation of the previously-mentioned components is beyond the scope of this book. Hence, readers are advised to go through official VMware documentation to develop more understanding of this.

Figure 12.1 shows the high-level overview of the Google Cloud VMware Engine along with its connectivity with the On-prem system:

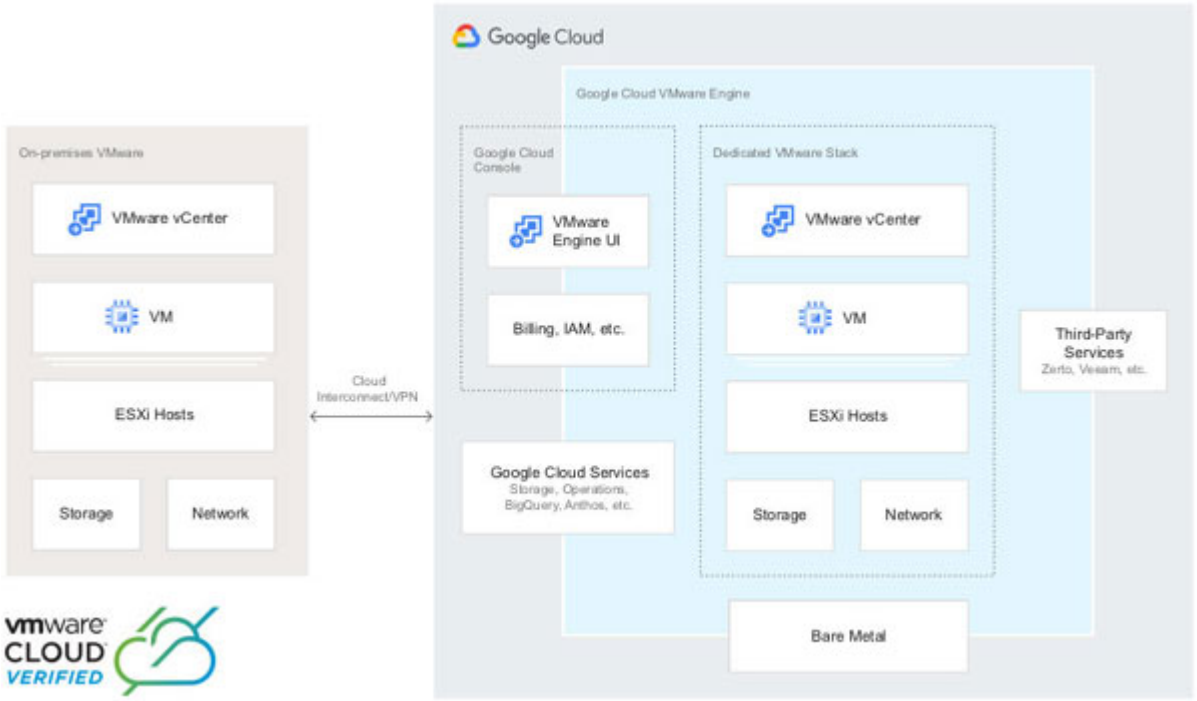

Figure 12.1: *Google Cloud VMware engine with On-Prem system.*
Source: *https://cloud.google.com/vmware-engine*

Features

Some of the important features of the Google Cloud VMware Engine are as follows:

- **Integration with GCP services**: Google cloud VMware Engine provides easy integration with GCP native services such as Cloud Storage, Google Cloud Operations Suite, and so on.
- **Monitoring and support**: Google cloud VMware engine is provided as **Infrastructure as a Service (IaaS)**. Hence, any issues with the underlying hardware, platform maintenance, upgrades, backup, monitoring and remediation, and so on are taken care of by Google itself.
- **Low latency network**: Google cloud VMware Engine is built on Google's low latency and high-performance network that supports 100Gbps bandwidth.
- **Hybrid connectivity**: Using networking services such as dedicated interconnect or Cloud VPN, you can connect the VMware Engine with

your On-Prem data center.

- **Easy management**: Many of the operational tasks, such as infra management, software upgrades, and so on, are taken care by Google itself. Therefore, you can focus more on business.

Oracle—Bare Metal Solution

Oracle Bare Metal Solution is an offering from Google Cloud to move your Oracle-based workloads to Google Cloud with almost no transformational requirement. It allows you to run your oracle databases the way you were running them On-prem. This solution offers you Oracle-certified hardware, where you can run your **Real Application Cluster** (**RAC**) with Oracle Data Guard for disaster recovery and RMAN for backup. It allows you to use your existing Oracle licensing, documentation, runbooks, and skillsets, with additional benefits of support, billing, and **Service Line Agreement**(**SLA**) offered by Google cloud. It comes with cascade bare metal servers, NVME storage, and Google Cloud 100G backbone network to run your Oracle workload.

Figure 12.2 depicts a generic overview of Google cloud bare metal solution:

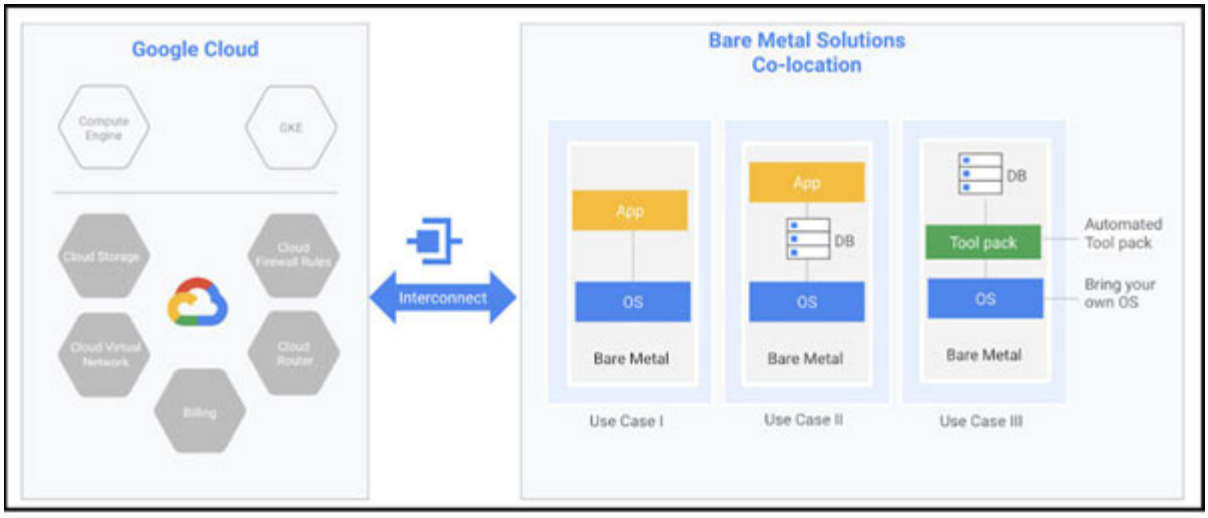

Figure 12.2: *Google cloud bare metal solution*
Source: *https://cloud.google.com/bare-metal*

Features

Following are some of the key features of Oracle-Bare Metal solution:

- **Infrastructure Support**: Google provides required infrastructure support with predefined SLAs. It provides 24*7 support for Severity 1 and Severity 2 issues.
- **Centralized Billing**: It provides centralized billing for Google Cloud and Bare Metal Solution.
- **Seamless use of Oracle native capabilities**: This solution comes with Oracle native capabilities such as Oracle RAC, RMAN, and Data Guard for backup and DR.
- **Security and protection**: This solution meets with the industry security and compliance requirements such as HIPPA, ISO, PCI DSS, and so on.
- **Certified Hardware**: Oracle has always been known for the licensing issue as there it needs to run only on certified hardware. Bare Metal solution comes with certified hardware that helps in eliminating any licensing issues.

Note: A detailed explanation of Oracle products and services is beyond the scope of this book. Therefore, readers are advised to go through official Oracle documentation to develop more understanding of this.

Anthos

Anthos is one of the most popular offerings of GCP, which is used to manage Kubernetes workloads in a hybrid cloud environment. It is a managed platform that lets you run and manage your application workloads across on-prem and multiple public clouds. Using Anthos, you can centrally manage these workloads. Anthos has lots of features that make it popular with customers who are looking for a hybrid or multi-cloud approach for deploying their workloads. Using Anthos, you can deploy, manage, and migrate microservices across various environments.

Anthos components

The following are the major components in Anthos:

- **Anthos clusters**: This is the primary computing environment for running Anthos that extends to GKEclusters in Google cloud, On-Prem, or multi-cloud environment. The Control plane in an Anthos cluster is responsible for managing the lifecycle of a Kubernetes cluster and is also responsible for registering and un-registering managed Kubernetes clusters in a variety of environments, including On-Prem, GCP, AWS, and Azure clouds.
- **Anthos Service Mesh**: Service Mesh is used to connect, manage, secure, and monitor microservices. It provides traffic routing and control, logs traces and metrics collection, and secure communication across different microservices. It also facilitates canary or A/B deployment of applications. *Anthos Service Mesh* is powered by *Istio*. Istio is an open-source platform that provides a secure way of connecting microservices.
- **Anthos config management**: Config manager is an important component of Anthos that provides a centralized way of pushing configuration to all the clusters. It also helps with the enforcement of policies on your cluster that prevent any configuration changes to Kubernetes API from violating any security and compliance controls. Anthos config manager stores configuration information in a centralized Git repository. Anthos config agent running on each cluster monitors changes in the cluster. If any change is detected, the agent will bring the cluster back to the previous state.
- **Binary Authorization**: Binary authorization is one of the services that enforce deploy-time security controls to ensure that only trusted container images are deployed in a GKE cluster.
- **Migrate for Anthos**: Migrate for Anthos is a tool that containerizes and migrates applications running on VMs to the GKE cluster.
- **Cloud run for Anthos**: Cloud Run for Anthos provides a developer-focused experience for creating modern applications on the Anthos platform that abstracts away the complexities of the underlying platform. It is based on Knative and provides a serverless, developer-focused experience.
- **Multi-cluster Ingress**: Ingress for Anthos is an entry point for microservices-based workload running on the GKE cluster.

- **Unified dashboard**: Anthos comes with a unified dashboard that gives you a unified view of application and Anthos resources. It shows you the status of microservices using service mesh and also provides you with the health status of your cluster.

Figure 12.3 shows you the Anthos dashboard in the Google Cloud console:

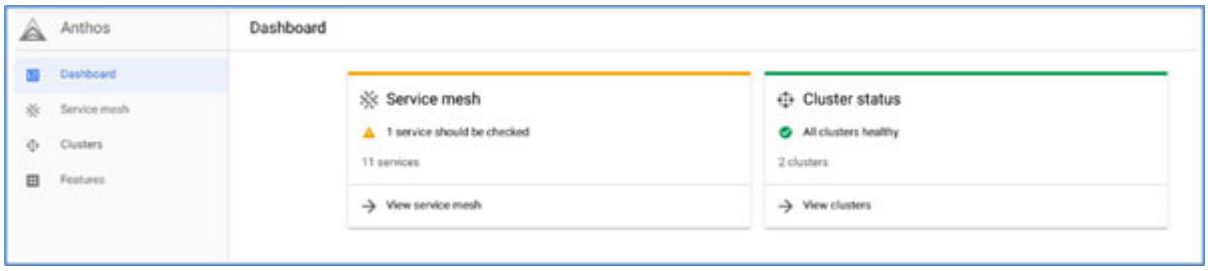

Figure 12.3: *Anthos Dashboard*
Source: *https://cloud.google.com/anthos/docs/concepts/overview*

Anthos features

Following are some of the important features of Anthos:

- **Container management and orchestration**: Anthos is a container management and orchestration service that helps to manage your containerized workload running on the GKE cluster. It can run on your existing VM environment but also comes with a bare metal option.
- **Centralized policy control**: Through config management, you can have centralized policy enforcement across various GKE clusters.
- **Modernizing existing workload**: Using **migrate for Anthos,** you can convert your existing non-containerized workload into containerized workload and move them to the cloud.
- **Traffic management**: Anthos service mesh comes with a fully managed traffic control plan, called traffic director. This helps you with deploying load balancers across clusters and VM instances. It also helps you with implementing traffic control policies.
- **Monitoring**: Using the Google Cloud operations suite, you can monitor the health and performance of the applications.
- **Security control**: Anthos comes with various security controls that can be added to each stage of the application life cycle, that is, development, build, and deployment.

Figure 12.4 represents Anthos and some of its features:

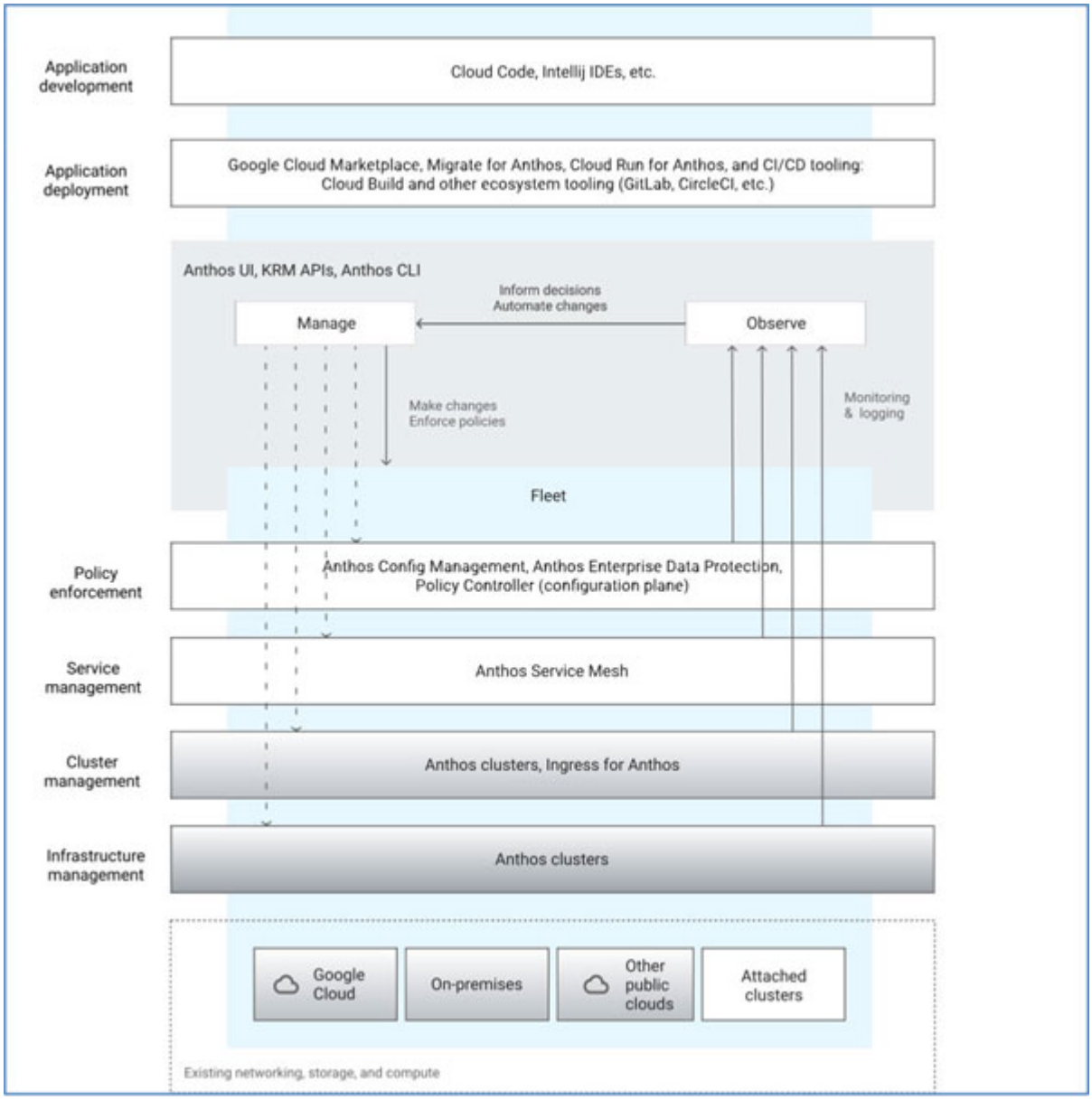

Figure 12.4: *Anthos components*
Source*: https://cloud.google.com/anthos/docs/concepts/overview*

Google Distributed Cloud Edge

Let us now briefly discuss about Google Distributed Cloud Edge offering.

Introduction to Edge Computing

Edge computing can be defined as a computing model in which data is captured, processed, and analyzed near the source of data. This helps with increasing the response time. It also helps with creating a near real-time experience due to greater speed and volume. Some examples of applications that use edge computing are self-driving cars, video games, and 5G network.

The following are some important components of edge computing:

- **Edge device**: Edge devices are the ones that collect and process the data locally. They have limited computing capacity. Some examples of Edge devices are IoT devices, smart watches, sensors, and so on.
- **Edge node**: These are the devices where actual processing happens for edge computing.
- **Cloud**: A public or private cloud is used to run the applications that manage the edge nodes.

Figure 12.5 depicts a high-level view of edge computing:

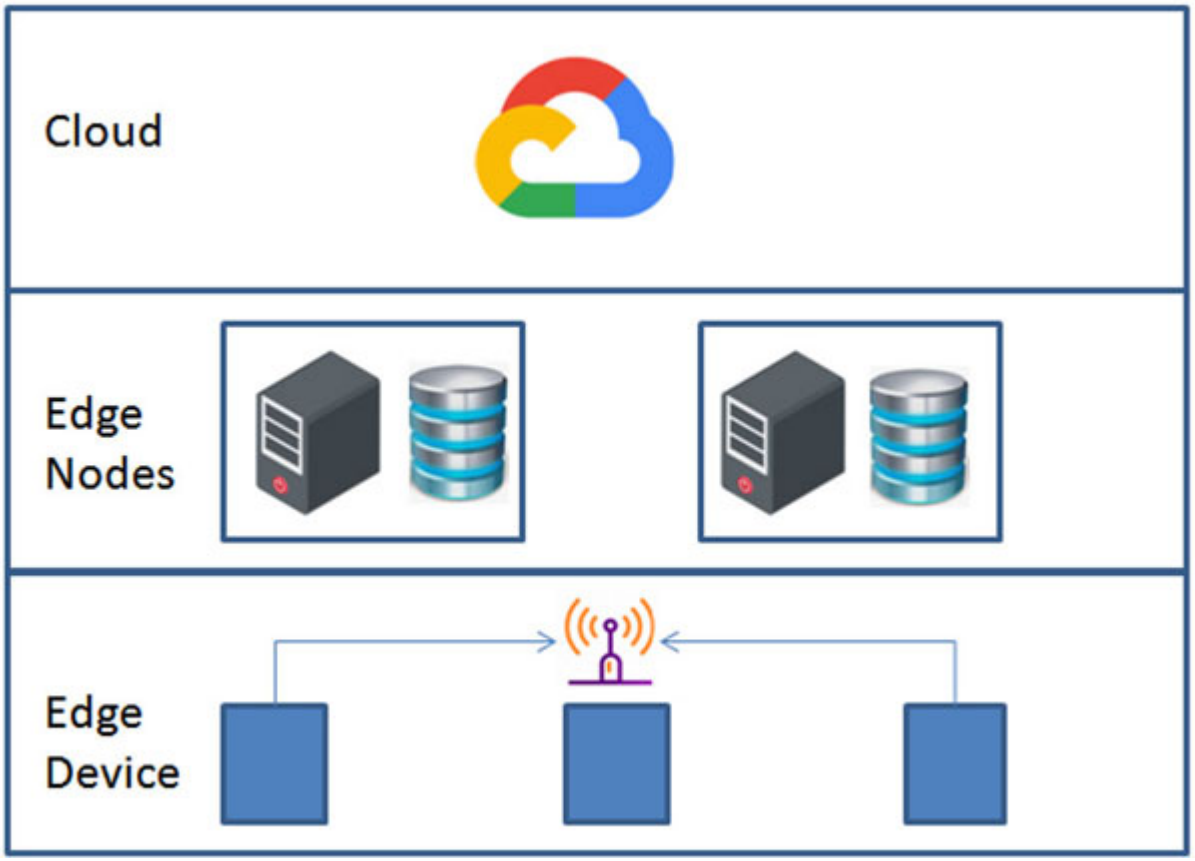

Figure 12.5: *Edge compute*

Distributed Cloud Edge

Google Cloud Distributed Cloud Edge is an offering from Google that brings Google cloud infrastructure and services to the edge locations. This offering is suitable for running a low-latency workload that runs closer to the customer's data center. This offering is suitable for running 5G core and RAN functions at the edge locations. Using Anthos, you can develop and deploy mission-critical applications on distributed cloud edge. It makes use of Intel and NVIDIA technologies to deliver the required performance to run the workload at the edge locations. Distributed Cloud Edge comes with GPU-accelerated computing and networking solutions.

The following are the locations where Google Distributed Cloud Edge can run:

- **Google's Network edge**: Google provides more than 140 edge locations that can be leveraged for deploying distributed cloud edge.
- **Customer Edge**: When due to compliance issues, data cannot move to Google's network edge; Distributed Cloud Edge can be run in customer's edge locations such as branch offices, retail stores, and so on.
- **Customer data center**: Can be deployed and run-on customer's data center considering their privacy, security, and compliance needs. This offering is called **Google Distributed Cloud Hosted**.
- **Operator edge**: Here, you can take advantage of telecommunication providers' edge locations.

Figure 12.6 features the interface of the Google Distributed cloud:

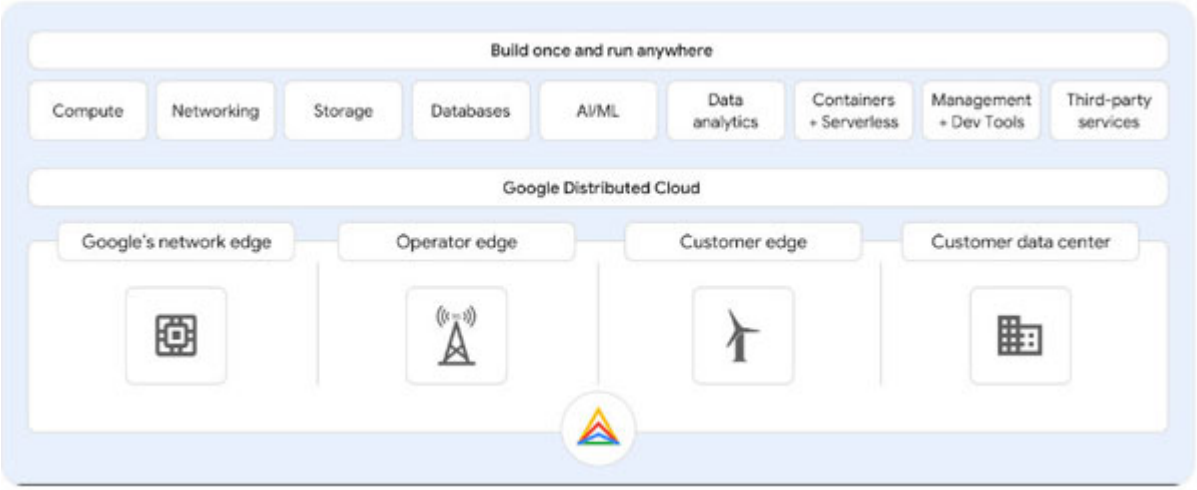

Figure 12.6: *Google distributed cloud*
Source*: https://cloud.google.com/blog/topics/hybrid-cloud/announcing-google-distributed-cloud-edge-and-hosted*

Google Cloud Backup and DR

Google Cloud Backup and DR service is the latest offering from Google Cloud Platform that provides centralized management of Backup and DR for Google cloud and hybrid workload. This Backup and DR Service fulfills low RPO and RTO requirements using intelligent data capture and recovery methods. Using this, service customers can use low-cost cloud storage for long-term retention and DR.

Figure 12.7 features Google cloud backup and DR dashboard:

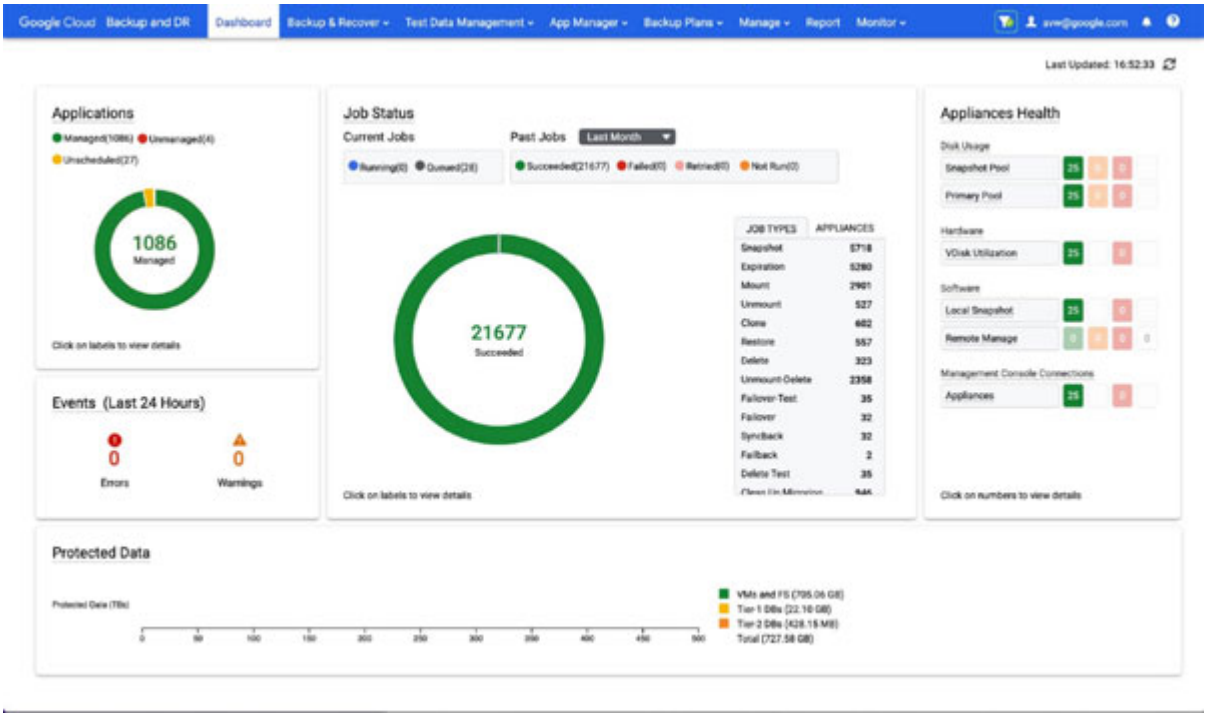

Figure 12.7: *Google Cloud backup and DR dashboard*
Source*: https://cloud.google.com/blog/products/storage-data-transfer/introducing-google-cloud-backup-and-dr*

Key features

Following are some of the key features of Google cloud backup and DR service:

- **Centralized management console**: Google cloud Backup and DR service come with a centralized management console that is used to configure, manage, and monitor your backups.
- **Incremental-forever backup**: This service provides you incremental-forever backup using **Change Block Tracking** (**CBT**) feature that helps you in optimizing space and bandwidth consumption with low RPO.
- **Agentless**: There is no need to install any agent on the VMware Engine and Compute Engine instances. In case if we need additional application protection, we require to install agents.
- **Instant Mount**: You can instantly mount your Google Cloud VMs and databases from a backup stored in Cloud Storage, which helps you with achieving a low recovery time objective.

- **Support for a variety of databases**: Using this service, you can take backups of a variety of databases such as SAP HANA, Microsoft SQL Server, Oracle, PostgreSQL, DB2, MySQL, MongoDB, and so on.
- **Cross-region disaster recovery**: You can recover your workload in a different region.

Figure 12.8 depicts the migration approach and targets for various kinds of workloads in Backup and DR service. For compute engine VMs, backups are stored as persistent disk snapshots, whereas in the case of VMware VMs, the backups target can be a persistent disk or cloud storage.

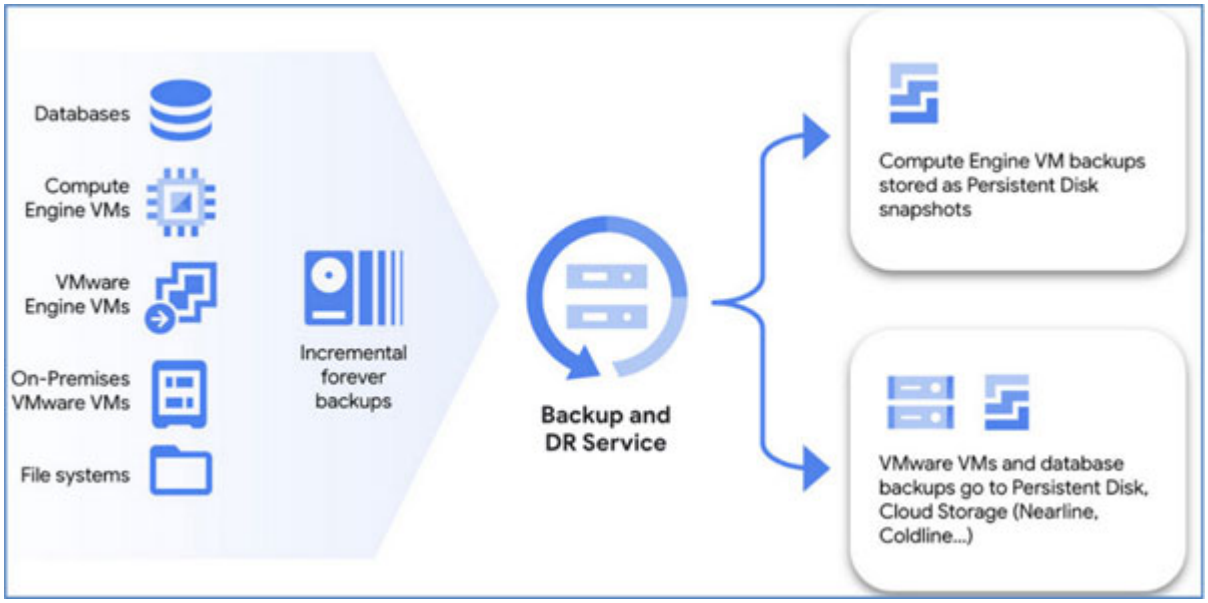

Figure 12.8: *Backup and DR service targets*
Source: *https://cloud.google.com/blog/products/storage-data-transfer/introducing-google-cloud-backup-and-dr*

Conclusion

In this chapter, we have discussed about a few additional GCP services available in the Google Cloud Platform. Please note that GCP is evolving very fast, and new GCP services are being introduced on a regular basis. In order to make one aware of these services, readers are advised to open a GCP account and subscribe to their notification service. Google will send a notification whenever any major service is introduced. In the upcoming and final chapter of this book, we will cover a few case studies that will help us

with developing some understanding of creating solutions for various requirements.

Key terms

- **Oracle real application cluster (RAC)**: It is a cluster using which you can run one single Oracle Database across multiple servers, resulting in high availability and fault tolerance.
- **RMAN**: RMAN stands for recovery manager. It is a backup utility that is built into the Oracle database itself and is used for performing backup and recovery operations of Oracle DB.
- **DataGuard**: It is a standby data database solution for Oracle that is used for disaster recovery and high availability.
- **CBT**: It is a technology that helps in creating incremental backup by tracking and copying a block of data that has seen changes**.**
- **NVDIA**: It is a technology company that is famous for producing **Graphics Processing Unit(GPU**), which are used in gaming, video editing, machine learning, and so on.

Questions

1. What is Google Cloud VMware Engine? Explain its important features.
2. Please explain Anthos and some of its features.
3. What is the purpose of using Google Distributed Cloud Edge service in GCP?

Further readings

- **https://cloud.google.com/vmware-engine**
- **https://cloud.google.com/bare-metal**
- **https://cloud.google.com/anthos**
- **https://cloud.google.com/distributed-cloud**
- **https://cloud.google.com/blog/products/storage-data-transfer/introducing-google-cloud-backup-and-dr**

CHAPTER 13

Use Cases

Introduction

In this chapter, we are going to discuss a few case studies that will cover many of the GCP services. We will take examples of some imaginary companies to discuss scenarios and possible solutions at a very high level.

Note: These are just case studies and in no way cover end-to-end solutions. It requires lots of client interactions and analysis to come up with the overall solution.

Structure

We will cover the following case studies:

- Ecommerce website deployment- XYZ limited
- Data transformation—ABC energy
- Landing Zone—AlphaBeta Limited

Objectives

By the end of this chapter, the readers will be able to make a decision on selecting appropriate services for similar kinds of requirements. This chapter will also help with building a strong foundation for creating solutions on GCP.

Ecommerce website deployment—XYZ limited

Let us consider a problem statement.

Problem statement

XYZ limited is looking to develop an e-commerce website that will be deployed on the Google cloud platform. This website will be developed in Java springboot and is supposed to have a microservices-based architecture, with MySQL as the backend database. The website will be accessed publicly by people across the globe and will have Prod and DR environment. The website should have an analytics component as well, which requires understanding the buying pattern of consumers/users by using data visualization tools. The solution should get implemented using automation tools and should also fulfill the continuous integration, development, and deployment requirement of the application. Apart from this, prevention from any DDoS attack and faster access tostatic content across the globe is expected.

Solution considerations

First, we will do a little brainstorming and try to develop a high-level solution approach. Here are a few pointers:

- Global VPC and IaaS and PaaS service of GCP considering global accessibility requirements.
- Multiregional replication will be required for the DR.
- As the application is internet-facing, we will require some external firewall as well.
- Global Load Balancer, considering multiregional approach across production and DR environment.
- Data analytics and visualization services for predictive analysis.
- Monitoring, IAM, and DNS services will be required.
- DevOps tools and services for application deployment and infrastructure provisioning.
- Compute, storage, and database services for running application workload.

GCP services to be used

On the basis of our overall solution approach, we can consider using the following major services for this requirement:

- Global Load balancer.
- GKE cluster for running the micro-services as containers. One can also consider using cloud run for a serverless approach, which requires no overhead of building and managing the clusters.
- CloudSQL for running database services.
- BigQuery for running analytics.
- Looker Studio for visualization.
- Google Cloud Operations Suite for monitoring and logging.
- Cloud Armor for Layer 7 firewall and DDoS protection.
- Cloud DNS
- Cloud IAM for identity and access management.
- Cloud CDN service for Global caching.
- KMS and Secret Manager for managing secrets and keys.
- Artifact registry for storing containerized images and artifacts.
- Cloud build, Terraform, Jenkins for DevOps implementation.
- Cloud storage for storing images or other static content and backup.

Deployment approach

Figure 13.1 represents a high-level deployment approach for the requirement. When the user makes an initial request, Cloud DNS works as a DNS resolver and converts it to the global any cast IP address. Cloud Armor works as a firewall for Layer 7 protection and also prevents it from DDoS attacks. Once the request has been validated, it is sent to the Webserver that captures any static Webpage requests from cloud CDN. Cloud CDN also stores this static data in it for any future requests. Web server interacts with the app server that resides on the GKE cluster. GKE cluster then processes the request using application logic and gets the required response from the database that resides in CloudSQL. This response is sent back to the Web server and finally to the end user. BigQuery data warehouse service is used to perform any analytics. Data studio integrates with BigQuery used for visualization and generating reports.

Refer to *figure 13.1*:

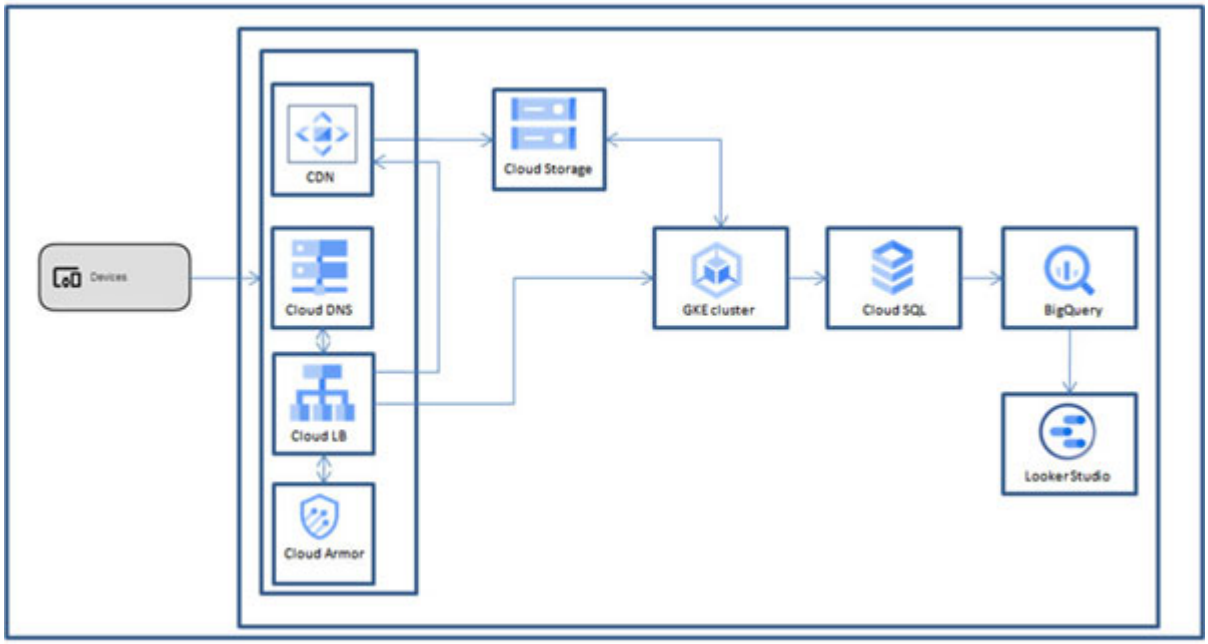

Figure 13.1: *Deploymentapproach*

Data transformation—ABC Energy

Let us consider a problem statement.

Problem statement

ABC Energy Limited is an energy company based out of the USA. The company has 1 million consumers across the country. There are various touch points available for the customer, namely, websites, mobile applications, WhatsApp, chatbots, and so on, to reach out for various services such as online view and download bills, payments, consumptions, and so on. Customers are influenced by the personalized services offered in the market for similar requirement, and hence, they are looking for improved and personalized services as per their needs. They are looking for a consultative approach concerning power usage. Residential and commercial customers are looking for their personal profiling that results in improved services. For meeting this need, ABC Energy Limited is looking for a data-like implementation that works as a single source of truth for their business. Data coming from various sources should be transformed, analyzed, visualized, and used for providing valuable insights using business intelligence.

Data sources

There are mainly two kinds of data sources identified for this requirement:

1. Data coming from data sources, such as On-Prem SAP and CRM systems, which have customer-centric data already available internally.
2. Data from specific social media platforms, such as Facebook, Twitter, Instagram, and YouTube, which can be captured through public APIs.
3. Realtime unstructured data are coming from the SCADA system through sensors. This includes streaming data that is being generated through various devices.

Considering the preceding problem statement and the data sources mentioned, we will try to create a proposed solution approach.

Solution considerations

Following are some of the important pointers that can be considered while defining a solution approach:

- Integration requirement with additional data sources, that is, On-Prem SAP and CRM system through connectors available in Dataflow.
- Historical data migration that is available On-Prem.
- Data ingestion, cleansing, transformation, analysis, and visualization need to be considered.
- Data modeling through visualization tools.

GCP services to be used

Following are the main GCP services that we have shortlisted on the basis of the previously-mentioned pointers for this requirement:

- IoT core for the data coming from external sensors via the SCADA system.
- Cloud Pub/Sub for event notification and messaging for streaming inputs.
- Cloud Dataflow for data transformation.

- Cloud Storage for storing any raw data for staging.
- BigQuery for data analytics.
- GCP Data studio and Looker for data visualization.
- Cloud Operations Suite for monitoring and logging.

Deployment approach

Figure 13.2 represents the high-level deployment architecture for the requirement. We have segregated the overall solution into various stages and shortlisted specific services that will fulfill a specific need:

1. One of the data sources is the SCADA system and sensors that will generate Realtime data. We will also have an SAP/CRM system from where data will be stored in source storage. This data will require a batch transfer. The data that will be coming through social media sites such as Facebook, Twitter, and so on will be accessed through respective APIs.
2. IoT core, Cloud Dataflow, or Pub/Sub will work as ingestion services.
3. Batch data and raw data will be moved to a staging area in cloud storage. Cloud Pub/Sub will be used as a messaging service for real-time ingestion of data to Cloud Dataflow. Dataflow will perform the required cleansing and transformation of the data, which will then move to BigQuery, to perform analytics.
4. SAP and CRM data will also go to cloud storage. Dataflow will be used for moving it to Cloud Storage, from where it will be moved to BigQuery as a dataset.
5. BigQuery integrates with Looker Studio, where graphs and dashboards can be created for BI activities.

Refer to *figure 13.2*:

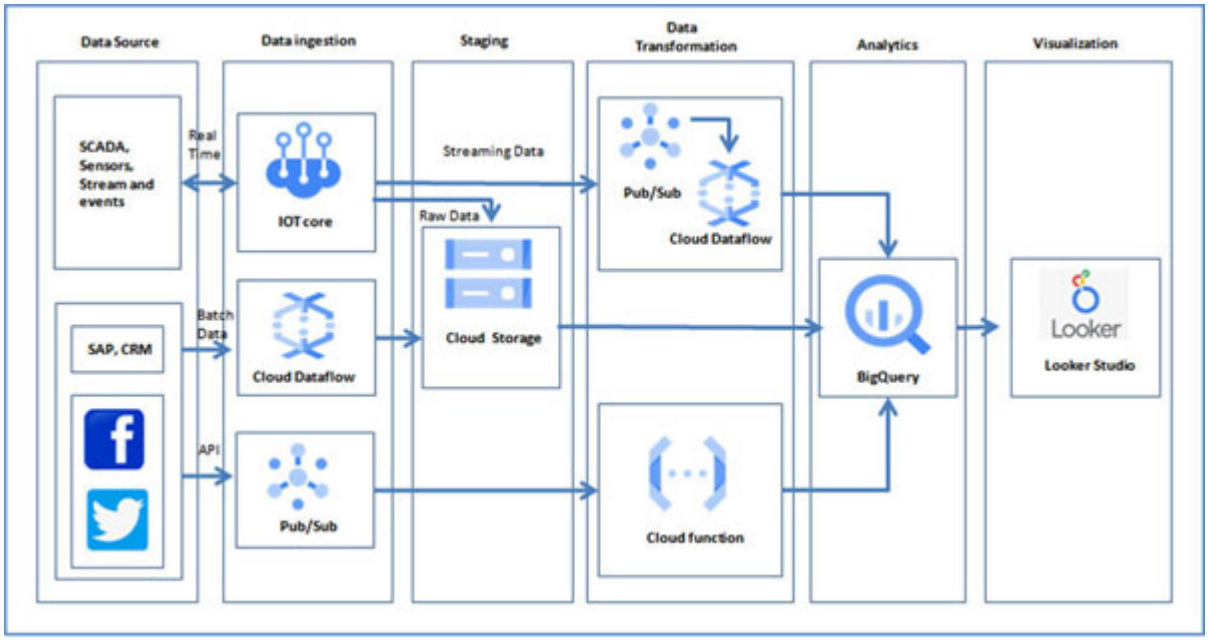

Figure 13.2: *Deploymentapproach*

Landing Zone—AlphaBeta Limited

Let us consider a problem statement.

Problem statement

AlphaBeta Limited is a vehicle manufacturing company looking to move the majority of its application workloads running on an On-Prem environment to the Google Cloud Platform. Their application workload consists of custom-built e-commerce and CRM applications. These applications are monolith applications. Some applications will still keep on running On-Prem due to compliance requirements, but they should get integrated with the applications running on Google cloud. AlphaBeta currently has Production, Non-Production, and Staging environments and wishes to maintain the same in Cloud. The requirement is to set a target landing zone in Google cloud where they can move their workload at a later stage. They wish to have a centralized network, logging monitoring, and billing services in the target environment in order to have a better administrative control. AlphaBeta has only one domain name associated with it, and all of its business functions fall under that. These are going to

be internal applications that should get accessed by internal users. DevOps should be a combination of open-source and Google cloud-native tools.

Solution considerations

Following are some of the pointers that we can consider while coming up with the landing zone design:

- Centralized control over Billing, monitoring, network, and security.
- Connectivity requirement with On-Prem data center, considering migration and integration needs.
- Production, Non-Prod, and staging environment requirements.
- Infrastructure automation and CI/CD implementation.

Landing Zone modules

The following are the main modules that should be considered while designing the landing zone:

- **Resource hierarchy**: This is used to organize the resources hierarchically. It will allow setting policies at different levels of the resources' hierarchy. The following are the different components of the resource hierarchy:
    - Organization hierarchy
    - Folder structure
    - Naming convention
    - Tagging and labeling strategy
- **Billing**: Used to define who is set to pay for the resource consumption and is also used for creating alerts, billing exports, and monitoring. The following are the components:
    - Billing account integration
    - Budget alerts
    - Billing data export
    - Reports and dashboard

- **Networking**: Defines the following connectivity options:
  - Hub and Spoke/Shared VPC
  - Hybrid connectivity
  - Inter VPC connectivity
- **IAM**: Defines identity and access control policies used for various users.
  - Organization policies and constraints
  - Identity federation
  - IAM policies and access creation
- **Automation**: Used for defining automated infrastructure provisioning and Code integration and deployment.
  - DevOps pipelines
  - Infra as a code tools installation/integration
  - Code repository integration (GitHub/Artifact registry)

Deployment approach

Figure 13.3 represents the resource hierarchy for the proposed landing zone:

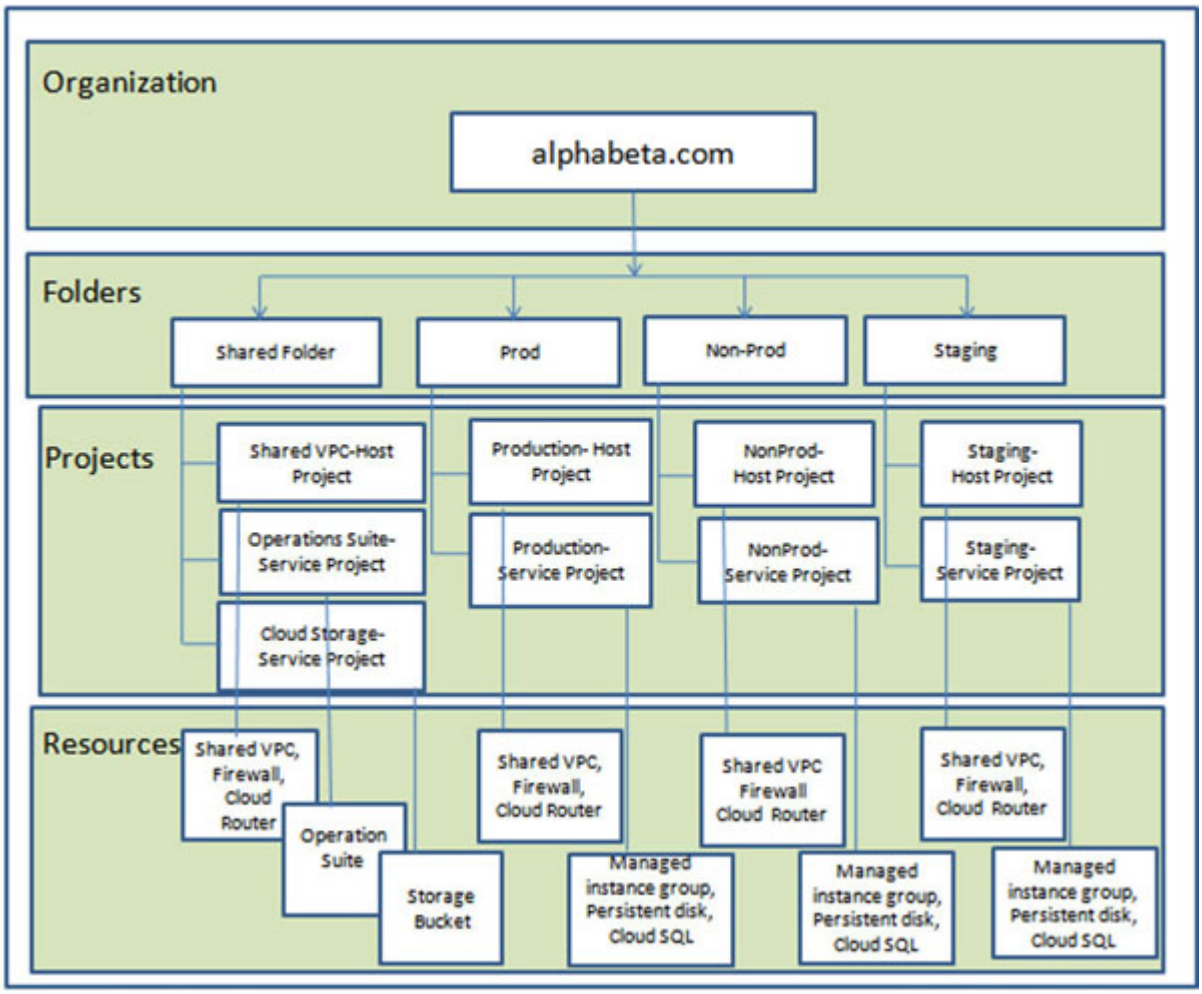

Figure 13.3: *Resource hierarchy*

Following are some of the pointers regarding the proposed resource hierarchy. Please note that the proposed landing zone hierarchy design is based on "*application environments*":

- **alphabeta.com** represents the organization in the resource hierarchy. This organization is tagged to the cloud identity account.
- Under organization, we have multiple folders. **Shared folder** includes all the projects that provide shared services to other projects/services.
- **Shared VPC Host Project** contains network and security services that will be shared with the Production, Non-Production, and Staging environment.
- **Shared VPC host project** is accessed by other services projects that are part of the same folder or other folders, that is, Production,

NonProd, staging, Cloud storage, and operations suite folders.

- **Production**, **Non-Prod**, and **Staging** folder will have their own host and service. While host projects will represent network and security services specific to their own projects, service projects will have their own resources. Host projects will have a network, subnets, and firewall rules configured, whereas Service projects will have to compute, storage, and database services configured.

Figure 13.4 represents the proposed landing zone architecture:

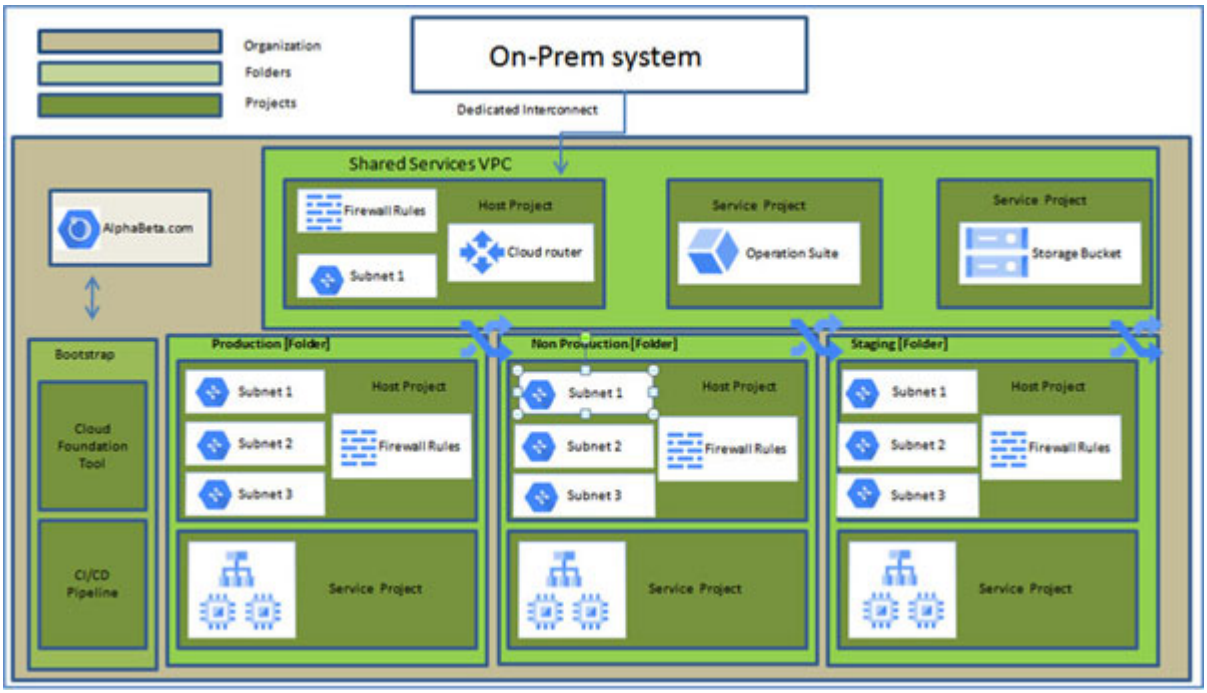

Figure 13.4: *Deployment approach*

Conclusion

In this chapter, we have discussed some important case studies that will be helpful for the readers in coming up with some realistic approaches while building various solution. Readers are advised to consider this as a starting point in their cloud journey. It is recommended to the readers to go through various case studies that are part of their certification program. There is a link that is shared at the end of this chapter. Readers are suggested to try building a high-level solution around those case studies by themselves.

Further readings

- **https://cloud.google.com/certification/guides/cloud-architect/casestudy-jencomart**
- **https://services.google.com/fh/files/blogs/master_case_study_ehr_healthcare.pdf**
- **https://services.google.com/fh/files/blogs/master_case_study_mountkirk_games.pdf**
- **https://services.google.com/fh/files/blogs/master_case_study_terramearth.pdf**

Index

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

R

S

T

U

V

W