

プログラマのための Google Cloud Platform 入門

阿佐 志保 [著]
中井 悦司 [著・監修]

お試し版

サービスの全体像から
クラウドネイティブ
アプリケーション構築まで

Google Cloud Platform の概要

アーキテクチャとコンセプト
開発／運用管理ツール

止まらない Web アプリケーション基盤の構築

Cloud Load Balancing
Cloud DNS

Web アプリケーション 実行基盤の構築

Compute Engine
Cloud Storage
Cloud SQL

コンテナ実行環境で マイクロサービス アーキテクチャを体験

Docker / Kubernetes
Container Engine
Container Registry
Cloud Datastore

機械学習を用いた GAE アプリケーション

App Engine
Cloud Datastore
Cloud Vision API
Cloud Translation API
Cloud Machine Learning Engine

- Stackdriver によるシステム運用
- Cloud IAM によるアカウント管理

Google Cloud Platform の概要

Google Cloud Platformは、Google社が提供するパブリッククラウドサービスです。このようなパブリッククラウドを活用して企業システムを構築する場合、それぞれのクラウドがもつ特徴を十分活かしたシステムアーキテクチャを採用することが重要になります。本章では、GCPの概要、そして、GCPがもたらすシステムアーキテクチャの変化を説明します。

◆ 1.1 Google Cloud Platformへようこそ

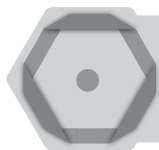
- 1.1.1 GCPがもたらすシステムアーキテクチャ
- 1.1.2 GCPのコンセプト

◆ 1.2 Google Cloud Platformを支えるインフラ

- 1.2.1 GCPの全体像
- 1.2.2 ユーザーアカウントとプロジェクト
- 1.2.3 仮想ネットワーク

◆ 1.3 はじめてのGoogle Cloud Platform

- 1.3.1 アカウント登録
- 1.3.2 プロジェクトの作成
- 1.3.3 Cloud Consoleの使い方
- 1.3.4 Google Cloud Platformの開発／運用管理ツール



1.1

Google Cloud Platformへ
ようこそ

Google Cloud Platform（以下、GCP）は、Google社が提供するパブリッククラウドサービスです。YouTubeやGmailなど、Google自身が提供するサービスを支えるインフラをベースに実装されているという特徴があります。Amazon社が提供するAmazon Web ServicesやMicrosoft社が提供するMicrosoft Azureと同様にグローバルにサービスを展開しており、数多くのシステムでの利用実績があります。

これらのパブリッククラウドを活用して企業システムを構築する場合、それぞれのクラウドサービスがもつ特徴を活かしたシステムアーキテクチャを採用することが、ユーザー企業のビジネスゴールに適合したシステムを構築するためのキーポイントとなります。また、従来のオンプレミスでのウォーターフォール型の開発手法とは異なる、クラウドネイティブな開発手法、あるいは、システムの構築／運用方法を導入することが、リリース期間の短縮やリリース後の安定稼働へとつながります。

本書では、GCPの特徴を活かしたシステムアーキテクチャを理解して、GCPを効果的に活用するための基礎を学びます。標準的なWebアプリケーション環境を出発点として、コンテナを利用したマイクロサービスアーキテクチャ、さらには、PaaSを用いた開発者にインフラの存在を意識させないアプリケーションアーキテクチャまでを段階的に理解していきます。それ

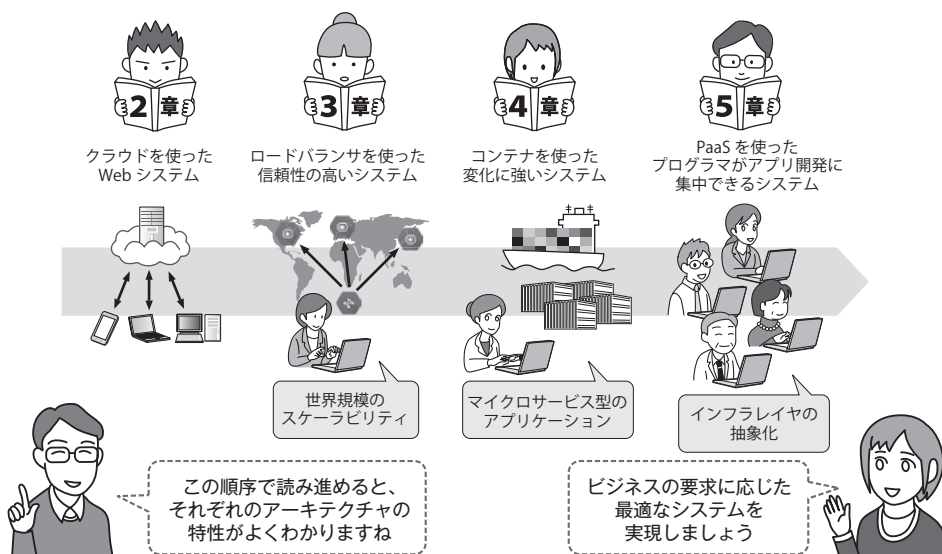


図 1.1 本書で説明するシステムアーキテクチャ

それぞれのアーキテクチャの特性を理解することで、ビジネスの要求に応じた最適なシステム構成を実現することが可能になります（図1.1）。

1.1.1 GCPがもたらすシステムアーキテクチャ

現在、Googleが提供するクラウドサービスは、「G Suite」や「Google Cloud Platform」などいくつかのブランドに分れています※1。G Suiteは、Gmail、Googleドライブ、Googleハングアウトなどのグループウェア（コラボレーションツール）をクラウド上で提供する、SaaSを中心としたサービスです。そして、本書の主題であるGCPは、ソフトウェアの開発、実行基盤をクラウド上で提供する、IaaSおよびPaaSを中心としたサービスになります。

GCPでは、さまざまなサービスが提供されています。特に、本書で取り上げる主なサービスは表1.1のとおりです。

表1.1 GCPが提供する主なサービス

カテゴリ	サービス	説明	関係する章
コンピューート	Compute Engine (GCE)	仮想マシンサービス	第2章、第3章
	Container Engine (GKE)	コンテナオーケストレーション	第4章
	App Engine (GAE)	運用が容易でスケーラブルなアプリケーションプラットフォーム	第5章
ストレージ／データベース	Cloud Storage	オブジェクトストレージ	第3章
	Cloud SQL	RDBMSのマネージドサービス	第2章、第3章
	Cloud Datastore	NoSQLデータベース	第4章
ネットワーキング	Cloud Virtual Network	仮想ネットワーク	第3章
	Cloud Load Balancing	ロードバランサ	第3章
	Cloud DNS	DNSのマネージドサービス	第3章
機械学習	Cloud Vision API	画像分析サービス	第5章
	Cloud Translation API	テキスト翻訳サービス	第5章
運用管理ツール	Cloud Console	統合管理コンソール	全章
	Cloud Shell	ブラウザ上で利用可能なコマンドラインコンソール	全章
	Stackdriver	統合監視サービス	付録
	Cloud IAM	ユーザーアクセス管理	付録

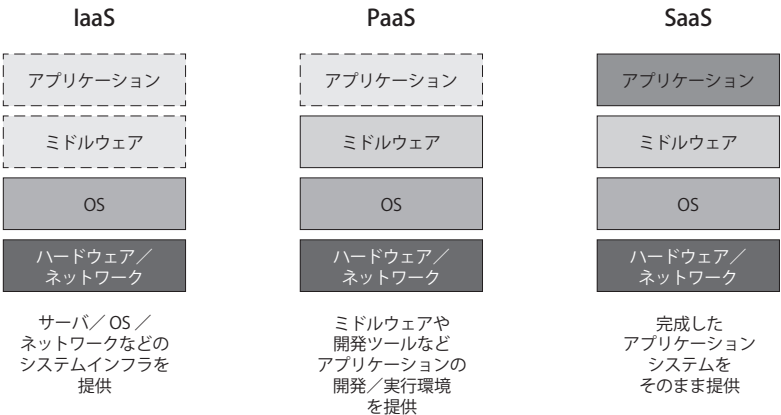
※1 最新の情報については以下の公式サイトを参照。

URL <https://www.google.com/cloud/>



クラウドサービスの種類

クラウドが提供する代表的なサービスは、それぞれが提供するサービスレイヤの違いにより、図Aのように分類されます。



図A クラウドサービスの種類

IaaS (Infrastructure as a Service)

サーバやOS／ネットワークなどのシステムインフラをオンデマンドに提供するサービスです。データを保存するストレージ領域も提供されます。サーバのスペックやデータ容量に加えて、信頼性や可用性などのサービスレベルに応じて利用料金が決まります。システムの要件に応じて、必要なサービスを組み合わせて利用します。

PaaS (Platform as a Service)

システムインフラに加えて、アプリケーションの開発に必要なミドルウェアをまとめて提供します。開発に必要なライブラリはもちろん、本番環境へのデプロイやシステム監視などの機能を提供しているものもあります。PaaS環境を利用することで、開発者は、システムインフラの構築／運用を気にせずに、アプリケーション開発に注力することが可能になります。

SaaS (Software as a Service)

すでにできあがったアプリケーション機能をサービスとして提供します。クラウドベンダーが提供するシステムをWeb経由で利用するという使い方が一般的です。ユーザーはアプリケーションを開発する必要がないので、要件に合致するサービスがあれば、迅速にシステムを利用できるのが特徴です。

この他には、スケーラブルなNoSQLデータベース（Key-Valueストア）であるCloud Bigtable、コンテンツ配信ネットワークを提供するCloud CDNなどのサービスがあります。本書では詳しく取り上げませんが、ビッグデータや機械学習についてのサービスが豊富に提供されているのもGCPの大きな特徴です（表1.2）。

表1.2 GCPが提供する主なビッグデータや機械学習サービス

カテゴリ	サービス	説明
ビッグデータ	BigQuery	大規模なデータ分析に対応したデータウェアハウス。ペタバイト規模のデータを低料金を格納して処理できる
	Cloud Bigtable	低レイテンシで高いスループットを実現するNoSQLデータベース。HBase互換APIを提供
	Cloud Spanner	ストロングコンシステンシーとスケーラビリティを兼ね備えた、分散型のリレーショナルデータベース
	Cloud Dataflow	ストリーミング処理とバッチ処理に対応した、フルマネージドのデータ処理サービス
	Cloud Dataproc	Spark／Hadoopのフルマネージドサービス
	Cloud Pub/Sub	リアルタイムで信頼性の高いメッセージングとデータのストリーミングサービス
機械学習	Cloud Vision API	画像分析サービスを提供するAPI。画像に含まれる情報を取得するだけでなく、不適切な画像を検出する機能や、文字認識(OCR) 機能を提供
	Cloud Speech API	音声のテキスト変換サービスを提供するAPI。高度なニューラルネットワークモデルを適用し、80以上の言語と方言を認識できる
	Cloud Translation API	テキスト翻訳サービスを提供するAPI。テキストをソース言語からターゲット言語に動的に翻訳できる。対応する言語は90種類以上
	Cloud Natural Language API	自然言語分析サービスを提供するAPI。構文解析や感情分析ができる
	Cloud Machine Learning Engine	TensorFlowを用いた機械学習サービス。GPUを利用して、大規模なモデルの構築／学習処理が実施可能

本書で扱うGCPのサービス

本書では、表1.1に示したGCPの主なサービスを組み合わせることで、アプリケーションの実行環境を構築していきます※2。特に、これからWeb技術を用いたアプリケーションの開発に関わっていくプログラマの方を対象として、GCPの基本的なサービスを活用しながら、アプリケーション開発に必要なインフラの基礎知識を解説していきます。GCPのクラウド上にWebアプリケーション環境を実際に構築することで、現代的なアプリケーション開発を支えるインフラの仕組み、さらには、効率的なWebアプリケーション開発の手法と考え方を身につけていただくことが本書の目標です。

第2章では、Google Compute Engine (GCE)、Cloud SQL、Cloud Storageを用いて、典型的なWebアプリケーション環境を構築することで、アプリケーション開発の基礎となるWeb技術を理解します。続く第3章では、Cloud Virtual NetworkとCloud Load Balancingを用いて、Webアプリケーション環境の冗長化を図り、アプリケーション開発者として知っておきたい、ネットワークインフラの仕組みを理解します。さらに、第4章では、Google Container Engine (GKE) と Cloud Datastoreを用いて、マイクロサービス型のアプリケーションをコンテナ環境にデプロイします。ここでは、開発したアプリケーションのデプロイやバージョンアップなど、アプリケーションのライフサイクルを通した管理の考え方を学びます。

そして、第5章では、Cloud Vision APIやCloud Translation APIなどの機械学習サービスを活用したアプリケーションの例を紹介します。ここでは、アプリケーションの実行環境として、PaaSタイプのサービスであるGoogle App Engine (GAE) を利用します。GAEでは、サーバ、ネットワークなどのインフラ、あるいは、アプリケーションのバージョン管理などの機能がすべてマネージドサービスとして提供されており、アプリケーション開発者自身は、これらを自分で用意する必要がなくなります。ただし、その裏側では、それまでに学んだネットワークやコンテナの技術が活用されています。GAEを利用する場合においても、このようなインフラの基礎知識が大切になります。

これらの章を通して学ぶのは、クラウド上のアプリケーション環境を実現するアーキテクチャの違いです。既存のオンプレミスシステムをクラウドに移行するだけではなく、物理的な制約を離れて、より効率的なアプリケーション開発を実現するには、クラウドならではのアーキテクチャを理解する必要があります。冒頭の図1.1に示したように、第2章～第5章で学ぶ内容は、次のようにまとめられます。

※2 最新の情報については以下の公式サイトを参照。

URL <https://cloud.google.com/products/>

- 第2章：標準的なWebアプリケーション環境を実現する方法
- 第3章：クラウドのネットワークインフラを用いて可用性の高いシステムを実現する方法
- 第4章：コンテナを利用してマイクロサービスアーキテクチャを実現する方法
- 第5章：PaaSを利用して、インフラを意識しない効率的なアプリケーション開発を実現する方法

Google App Engine と GCP の歴史

ここで過去の歴史をすこし振り返ると、2008年に公開されたGoogle App Engine (GAE)が、現在のGCPの起源となる最初のサービスとなります。開発したアプリケーションのプログラムコードをアップロード（デプロイ）すると、コンテナをベースとした独自の実行環境でコードの実行が開始されるという、いわゆるPaaSタイプのサービスです。バックエンドのデータベースには、Cloud Datastoreという独自の分散データストアが用意されています。GAEは、アプリケーション実行環境の構築／メンテナンスが不要なサービスで、利用者は、アプリケーションの開発に専念することができるというメリットがあります。そのため、専任のインフラエンジニアや運用担当者を確保することが難しいスタートアップ企業を中心に利用が広がりました。一例を挙げると、1日の利用者数が1億人を超えるとも言われるスマートフォンアプリ「Snapchat」のシステムはGAEを用いて構築されており、2014年の時点でたった25人の技術者により開発／運用が行われています※3。

その一方、GAEは、オンプレミスで構築されてきた従来型のサーバインフラとは考え方や利用方法が異なることもあり、オンプレミスからクラウドへの移行という目的でGAEを利用するユーザーはそれほど多くはありませんでした。

その後、2013年になると、仮想マシン環境を提供する、より汎用的なIaaSタイプのサービスとしてGoogle Compute Engine (GCE) の提供が開始され、翌年の2014年からは、MySQLのマネージドサービスである、Cloud SQLが利用可能になりました。これにより、仮想マシンとリレーショナルデータベースを組み合わせた伝統的なサーバ環境をクラウド上に用意できるようになり、オンプレミスのサーバ環境を比較的容易にGCPに移行できるようになりました。

ちなみに、最近では、**サーバレスアーキテクチャ**や**NoOps**など、サーバやOS環境を意識しないアプリケーション実行環境の考え方が広がってきました。GAEを始めとするGCPが提供するサービス群は、サービス開始当初より、アプリケーションの開発／実行に必要なコン

※3 [Google Cloud Platform Live: Interview with SnapChat's Bobby Murphy]

URL <https://www.youtube.com/watch?v=17PtS1Qx8kU>

ポーネントを抽象化されたAPIサービスとして提供するという考え方で設計されており、このような現代的なアプリケーションアーキテクチャの思想を先取りしたものでした。GCEが提供する仮想マシンを用いて環境を構築する場合でも、Cloud StorageやCloud SQLなど、適切なマネージドサービスを組み合わせることが、運用効率が高く、耐障害性に優れたアプリケーション環境を実現するポイントになります。

現在、GCPでは、GCEが提供する仮想マシン環境と、GAEが提供するPaaS環境の中間に位置するサービスとして、Google Container Engine (GKE) が提供されています。これは、オープンソースソフトウェアのKubernetesを用いた、コンテナオーケストレーションシステムを提供するもので、Dockerを始めとするコンテナ管理ツールの普及にともない、高い注目を集めるようになりました。本書では、GCEを用いた従来型のサーバ環境と、GKEを用いたコンテナ型のアプリケーション環境のそれぞれについて、基礎となる考え方と具体的な利用方法を詳しく解説しています。それぞれの特徴を理解して、アプリケーションの特性やシステム要件に応じて適切に使い分けていくことが、GCPを使いこなす上で1つのポイントになるでしょう。GCE、GKE、GAEの違いは、表1.3のようにまとめることができます。

表1.3 GCE、GKE、GAEの違い

	GCE (Google Compute Engine)	GKE (Google Container Engine)	GAE (Google App Engine)
提供される環境	仮想マシン	Kubernetesを用いたコンテナクラスタ	独自のアプリケーション実行環境
アプリケーションのデプロイ方法	ゲストOS上にアプリケーションをインストール	Dockerイメージからコンテナをデプロイ	アプリケーションコードをデプロイ
主な利用方法	仮想マシンを用いた従来型のアプリケーション実行環境を構築	コンテナによるマイクロサービスの実行環境を構築	スケーラビリティと開発効率が重視されるアプリケーションの開発／実行環境として使用

なお、GAEには、独自のコンテナインフラで動作するStandard Environmentと、仮想マシン上のDockerコンテナで動作するFlexible Environmentの2種類があります。Flexible Environmentのほうがカスタマイズの自由度は高くなりますが、高速なスケールアウトといった基本性能や運用管理の容易性は、Standard Environmentのほうが優れています^{※4}。

※4 本書では、Standard Environmentのみを取り扱います。GAEに関する説明は、すべてStandard Environmentを前提としたものになります。

1.1.2 GCPのコンセプト

前項でも触れたように、GCPが提供するサービスは、アプリケーションの開発／実行に必要なコンポーネントを抽象化されたAPIサービスとして提供するという思想があります。とりわけ、スケーラビリティが高く、世界規模で利用されるアプリケーションを比較的容易に実現できるという特徴があります。ここには、GCPで提供されるサービスは、Googleが自社内で利用するインフラやソフトウェア環境が基礎になっているという理由があります。

現在、Googleは、GmailやGoogleドライブなど、G Suiteに含まれるSaaSタイプのクラウドアプリケーション、あるいは、検索エンジン、YouTubeなどのコンシューマー向けサービスを自社が保有する世界各地のデータセンターとそれらを相互接続するグローバルなネットワーク回線を用いて提供しています。これらのサービスには、「世界のあらゆる人々を対象ユーザーとする」という大きな特徴があり、サービスを提供するアプリケーションは、全世界のデータセンターで共通に稼働する必要があります。そこで、Googleでは、世界中のデータセンターを共通の仕組みで標準化した上で、共有ストレージやデータベース、ロードバランサなど、アプリケーションの稼働に必要となるミドルウェアを独自のソフトウェア技術で実装してきました。アプリケーション開発者は、これらの機能をAPI経由でオンデマンドに利用しながら、アプリケーションの開発からテスト、そして実サービス環境へのデプロイまでを標準化されたプロセスにしたがって進めていきます。

よく考えると、これは、「クラウドを活用したDevOpsの実現」そのものではないでしょうか？ Googleの社内インフラは、地理的な制約をなるべく排除して、利用者数やデータ量の増加に対応したスケーラブルなアプリケーションを開発するための理想を追求した環境と言えます。このようなGoogle独自のインフラ環境をクラウドサービスとして、外部のユーザーにも開放したものがGCPだと考えると、その全体像がよりスッキリと理解できるかもしれません（図1.2）。もちろん、Google社内で利用されている環境をそのままの形で公開するのではなく、Cloud Consoleを始めとするユーザーインターフェースやCloud IAMによるユーザーアクセス制御など、パブリッククラウドとして、より使いやすい環境が整えられています。

Chap1

Chap2

Chap3

Chap4

Chap5

App

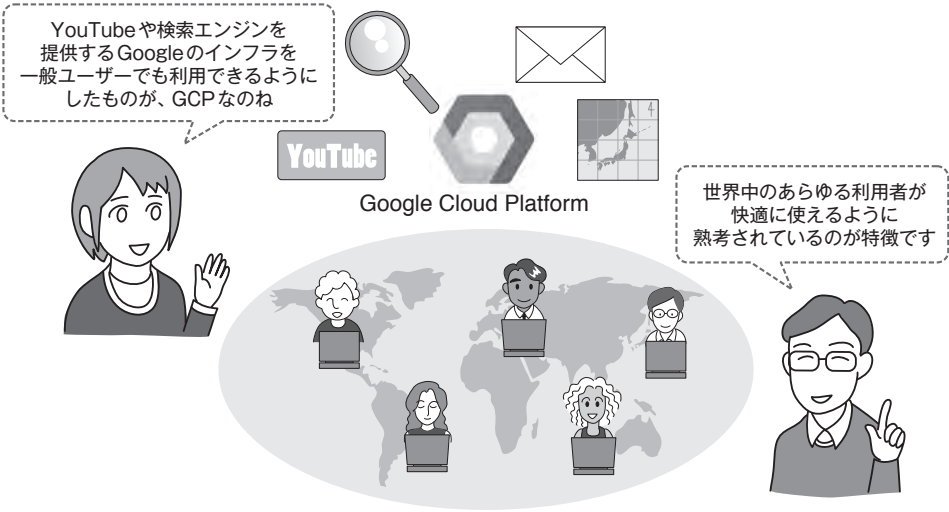


図 1.2 GCPのコンセプト

ちなみに、GCP、あるいは、Google の社内インフラで利用されているソフトウェアの中には、研究論文という形でアーキテクチャが公開されているものがあります。一般に、基盤ソフトウェアを使いこなす上では、「それが何を目的して開発されたものなのか」「どのようなユースケースに最適化された仕組みなのか」という点を正しく理解することが大切です。GCP を徹底的に使いこなしたいという方、あるいは、Google が独自開発した分散ソフトウェア技術に興味のある方は、これらの論文に目を通して、内部アーキテクチャという観点から、その技術的な特徴を捉えてみるのも面白いでしょう。表 1.4 は、参考として、代表的な研究論文をまとめたものになります。

表 1.4 GCPに関連するソフトウェアの研究論文（代表例）

論文タイトル／URL	ソフトウェア名称	関連する GCP サービス
Maglev: A Fast and Reliable Software Network Load Balancer URL http://research.google.com/pubs/pub44824.html	Maglev	Cloud Load Balancing
Bigtable: A Distributed Storage System for Structured Data URL http://research.google.com/archive/bigtable.html	Bigtable	Cloud Bigtable
Megastore: Providing Scalable, Highly Available Storage for Interactive Services URL http://research.google.com/pubs/pub36971.html	Megastore	Cloud Datastore

論文タイトル／URL	ソフトウェア 名称	関連する GCPサービス
Dremel: Interactive Analysis of Web-Scale Datasets URL http://research.google.com/pubs/pub36632.html	Dremel	BigQuery
Large-scale cluster management at Google with Borg URL http://research.google.com/pubs/pub43438.html	Borg	Container Engine
FlumeJava: Easy, Efficient Data-Parallel Pipelines URL http://research.google.com/pubs/pub35650.html	FlumeJava	Cloud Dataflow
MillWheel: Fault-Tolerant Stream Processing at Internet Scale URL http://research.google.com/pubs/pub41378.html	MillWheel	Cloud Dataflow
Spanner: Google's Globally-Distributed Database URL https://research.google.com/archive/spanner.html	Spanner	Cloud Spanner



1.2

Google Cloud Platformを支えるインフラ

GCPの利用を開始するにあたり、GCPの全体像を把握しておきましょう。ここでは、GCPを構成するデータセンターやグローバルネットワークの仕組み、プロジェクトとユーザーアカウントの関係などを整理しておきます。



1.2.1 GCPの全体像

本節では、GCPを構成するインフラの全体像を確認します。物理的な意味では、世界各地のデータセンターで構成されるリージョンとゾーン、そして、これらを相互接続するグローバルネットワークがあります。また、リソースの割り当てや課金先を管理するための論理的な枠組みとして、ユーザーアカウントとプロジェクトの考え方を理解しておく必要があります。これらの全体像は、図1.3のようにまとめることができます。

まず、GCPのサービスを利用する際は、リージョンとゾーンによってリソースを配置する場所が決まります。リージョンは、データセンターが存在する地域を表すもので、さらに、1つのリージョンの中に複数のゾーンが用意されています。それぞれのゾーンは、一般に「障害ドメイン (Failure domain)」と呼ばれる区画を構成するもので、複数のゾーンにまたがって

Chap1

Chap2

Chap3

Chap4

Chap5

App

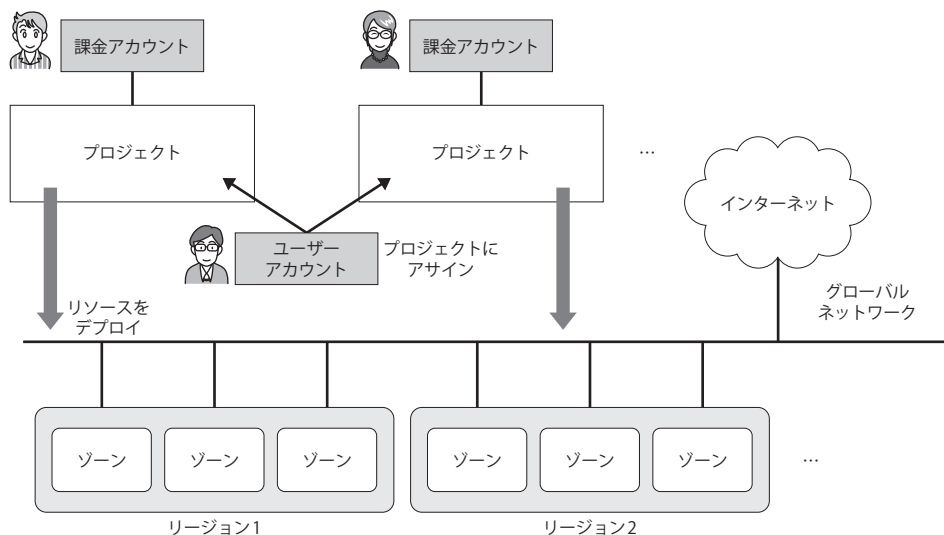


図1.3 GCPの全体像

リソースを配置することで、単一障害点を排除した冗長構成を実現することができます。サービスの種類によって、リージョンだけを指定すると、自動的に複数のゾーンにまたがった冗長構成になるものと、明示的にゾーンを指定して利用するものがあります。2017年4月時点では、全世界で7か所のリージョンが利用可能で、それぞれのリージョンには、2〜4個のゾーンが用意されています(図1.4)※5。各ゾーンは地域名と1文字の識別子を組み合わせた名前で識別されます。たとえば、東アジア地域のゾーンaは「asia-east1-a」と名付けられています。

さらに、それぞれのリージョンはGoogleが保有する内部ネットワークで相互接続されており、リージョン間においても、インターネット回線を経由することなく、高速な内部ネットワークを介した通信が可能です。実際にネットワークを使用する際は、1.2.3「仮想ネットワーク」で説明する仮想ネットワーク機能を利用して、プロジェクトごとに独立した仮想ネットワーク環境を構成して利用します。

インターネットとの接続については、全世界に70か所以上ある、PoP (Point of Presence) と呼ばれる相互接続地点を介して行われます。外部のユーザーがGCP上のリソースにアクセスする場合、アクセス元に応じた最適なPoPを経由して、内部ネットワークを通してアクセス先のリソースへと到達します。つまり、GCP上にアプリケーションをデプロイすれば、

※5 最新の情報については以下の公式サイトを参照。

URL <https://cloud.google.com/compute/docs/regions-zones/regions-zones>

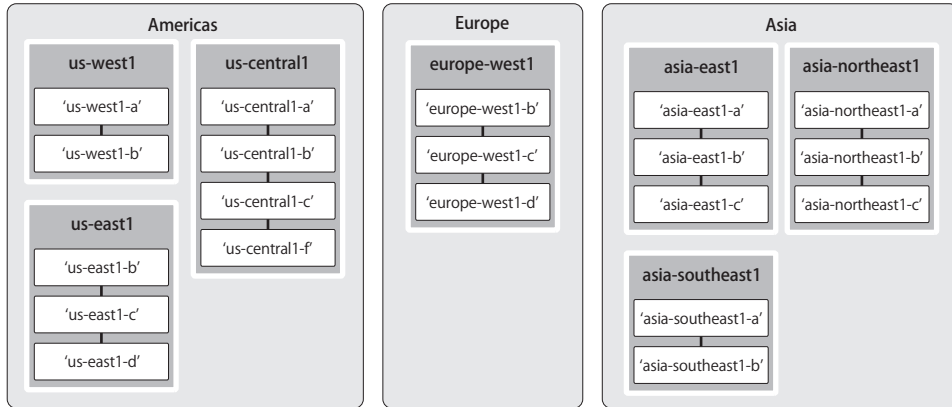


図1.4 GCPを構成するリージョンとゾーン

Googleが保有する内部ネットワークを利用して、全世界のユーザーにアプリケーションサービスを提供することが可能になるわけです。

1.2.2 ユーザーアカウントとプロジェクト

GCPを利用するユーザーは、Gmailなどで利用される **Google アカウント** を用いて認証を行います。事前を取得した Google アカウントを用いて GCP の利用登録を行うことで、GCP の利用を開始することができます。具体的な登録手順については、1.3「はじめての Google Cloud Platform」で説明します。

GCP のサービスを利用する際は、はじめに **プロジェクト** を作成して、課金アカウントとの紐付けを行います。一般に、個人で GCP を利用する場合は、プロジェクトを作成したユーザー自身を課金アカウントとして登録します。プロジェクトは、一般に **テナント** と呼ばれる概念に相当するもので、該当プロジェクトに登録されたユーザーアカウントのみが、プロジェクト内のリソースを操作することができます。プロジェクトを作成したユーザーアカウントの他に、他のユーザーアカウントやユーザーグループを追加登録して、「オーナー」「編集者」「閲覧者」などのアクセス権限を設定することができます。プロジェクト内で使用したリソースの利用料金は、先に設定した課金アカウントに対して請求が行われます。

1人のユーザーが複数のプロジェクトを作成することも可能で、それぞれのプロジェクトは完全に独立した環境になります。また、使用中のプロジェクトをシャットダウンすると、その中で利用していたリソースはすべて削除され、一切の課金は行われなくなります（正確には、

30日間の猶予期間の間は、シャットダウンを取り消して、元の状態に戻すことが可能です。そのため、一時的にテスト用のプロジェクトを作成して、テストが終われば、プロジェクトごとシャットダウンするなどの使い方も可能です。この場合、リソースを停止／削除し忘れて、無駄な課金が発生する心配ありません。

1.2.3 仮想ネットワーク

GCPでは、それぞれのプロジェクトにおいて、独立した仮想ネットワークを構成することが可能です。さらに、1つのプロジェクト内に複数の仮想ネットワークを定義することもできます。仮想ネットワークを定義する際は、**Legacy Network**、もしくは、**Subnet Network**のどちらかを選択します※6。Legacy Networkの場合、図1.5のように、すべてのリージョンにま

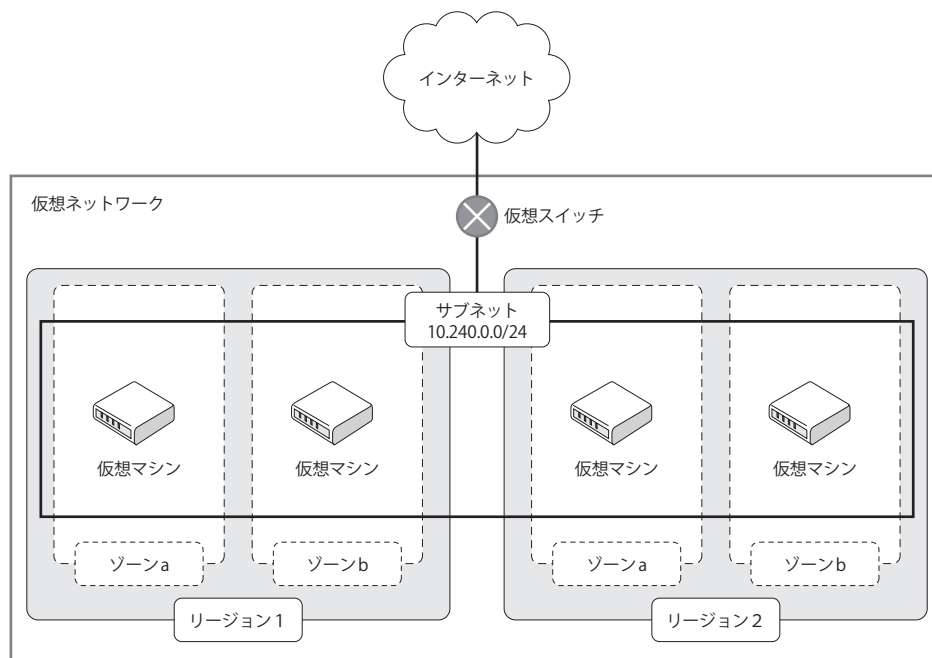


図1.5 Legacy Networkによる単一サブネット構成

※6 Legacy Networkは、gcloudコマンドを用いて作成する必要があります。Cloud ConsoleのGUI画面からは作成できないので注意してください。特別な要件がない場合は、Subnet Networkを使用するほうがよいでしょう。

たがった単一のサブネットが構成されます。一方、Subnet Networkでは、図 1.6 のように、リージョンごとに個別のサブネットが用意されます。複数の Subnet Network を定義することにより、1つのリージョンに複数のサブネットを割り当てることも可能です。

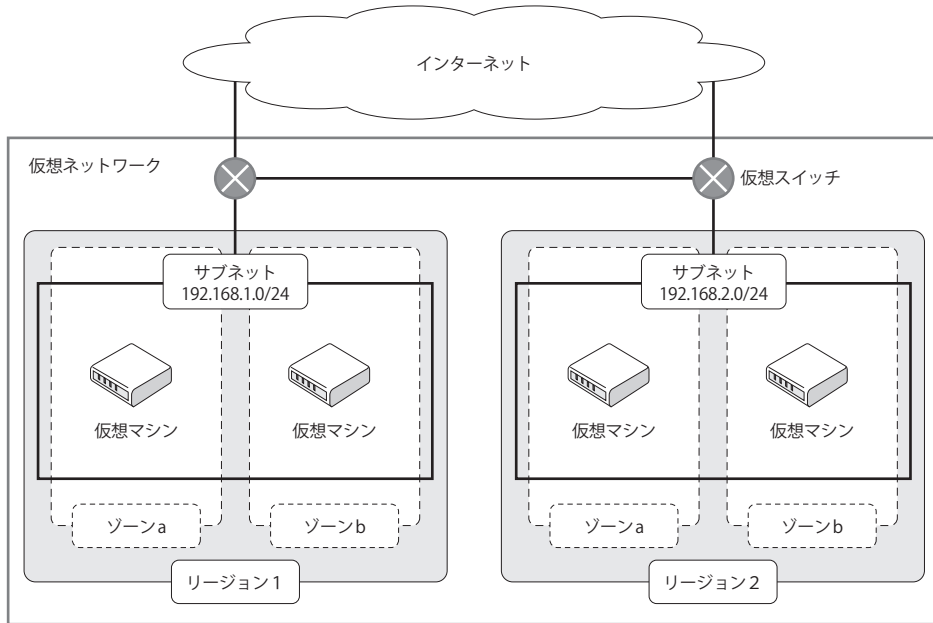


図 1.6 Subnet Networkによる複数サブネット構成

なお、3.2.3「GCPのファイアウォール機能」で説明するように、GCPの仮想ネットワークでは、仮想マシンに付与したタグによって、ファイアウォールの設定を行います。そのため、ネットワーク通信を分離する目的でサブネットを複数用意する必要はありません。Legacy Networkを使用した場合でも、ファイアウォールによるパケットフィルタリングを実施することは可能です。

この他には、ユーザー企業の社内ネットワークをGCPの内部ネットワークと直接に接続する、Cloud Interconnectというサービスも提供されています*7。

*7 [Cloud Interconnect]

URL <https://cloud.google.com/interconnect/>

続きは本書にて
お楽しみください

Amazonで購入

SEshopで購入

他書店で購入



著者・監修者

中井 悦司（なかい えつじ）

1971 年 4 月大阪生まれ。ノーベル物理学賞を本気で夢見て、理論物理学の研究に没頭する学生時代、大学受験教育に情熱を傾ける予備校講師の頃、そして、華麗なる(?) 転身を果たして、外資系ベンダーで Linux エンジニアを生業にするに至るまで、妙な縁が続いて、常に Unix/Linux サーバと人生を共にする。その後、Linux ディストリビューターのエバンジェリストを経て、現在は、米系 IT 企業の Cloud Solutions Architect として活動。最近では、機械学習をはじめとするデータ活用技術の基礎を世に広めるために、講演活動のほか、雑誌記事や書籍の執筆にも注力。主な著書は、『[[改訂新版] プロのための Linux システム構築・運用技術』『Docker 実践入門』『IT エンジニアのための機械学習理論入門』（いずれも技術評論社）、『TensorFlow で学ぶディープラーニング入門』（マイナビ出版）など。



著者

阿佐 志保（あさ しほ）

金融系シンクタンクなどで銀行／証券向けインフラエンジニア、製造業向けインフラエンジニアとして従事。都市銀行情報系基盤システム構築やシステム統合、証券会社向けバックオフィスシステムの共通基盤開発や統合認証基盤構築プロジェクト、石油／LNG プラント建設を行うエンジニアリング企業のシステム基盤構築プロジェクトなどを経験。出産で離職後、Linux やクラウドなどを独学で勉強し、技術書を執筆。約 8 年間の専業主婦を経て、2016 年 5 月より、TIS 株式会社の調査研究部門である戦略技術センターで、広報・人材採用を担当。主な著書は、『プログラマのための Docker 教科書』『Amazon Web Services ではじめる新米プログラマのためのクラウド超入門』（いずれも翔泳社）など。

装丁 森裕昌（森デザイン室）
本文デザイン・DTP 株式会社シックス

プログラマのための Google Cloud Platform 入門

サービスの全体像からクラウドネイティブアプリケーション構築まで

2017 年 6 月 1 日 初版第 1 刷発行

著 者 阿佐 志保（あさ しほ）
著者・監修 中井 悦司（なかい えつじ）
発行人 佐々木 幹夫
発行所 株式会社 翔泳社 (<http://www.shoeisha.co.jp>)
印刷・製本 株式会社 廣済堂

© 2017 Etsuji Nakai / Shiho Asa

※本書は著作権法上の保護を受けています。本書の一部または全部について（ソフトウェアおよびプログラムを含む）、株式会社 翔泳社から文書による許諾を得ずに、いかなる方法においても無断で複写、複製することは禁じられています。

※本書へのお問い合わせについては、ii ページに記載の内容をお読みください。

※落丁・乱丁はお取り替えいたします。03-5362-3705 までご連絡ください。

ISBN978-4-7981-3714-8

Printed in Japan