Global Knowledge®

Expert Reference Series of White Papers

Understanding Google Cloud Platform: Architecture

Understanding Google Cloud Platform

Alex Meade is a Global Knowledge instructor and DevOps consultant, currently focusing on Google Cloud Platform, DevOps and Docker containers

Google Cloud Platform (GCP) is Google's public cloud offering comparable to Amazon Web Services and Microsoft Azure. The difference is that GCP is built upon Google's massive, cutting-edge infrastructure that handles the traffic and workload of all Google users. As such, GCP has courted numerous customers that need to run at an enormous global scale, such as Coca-Cola and Niantic (creators of Pokémon Go). A detailed explanation of how Google helped Pokémon Go scale to 50x their expected traffic in just a few days after the game launched can be found here.

There is a wide range of services available in GCP ranging from Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) to completely managed Software-as-a-Service (SaaS). In this paper, we will discuss the available infrastructure components and how they provide a powerful and flexible foundation on which to build your applications.

Google Compute Engine (Virtual Machines)

Hosted virtual machines (VM) are usually the first thing people think of when moving to the cloud. While GCP has many ways to run your applications such as App Engine, Container Engine, or Cloud Functions, VMs are still the easiest transition to the cloud for architects and developers that are used to having complete control over their operating system.

Google Compute Engine (GCE) supports running both Linux and Windows virtual machines. Either Google-maintained machine images can be used to provision VMs or images from your existing infrastructure can be imported. One common practice is to provision an image for a particular OS, install all needed software dependencies for your applications, then create a new image from that VM. This gives you a "pre-baked" image that you can quickly deploy without having to wait for software to install before the VM can begin doing valuable work. Another strategy is to install a common set of tools into an image, such as a company-wide compliance package, then create a "golden" image that you share with development teams to use for their applications.

There are a number of cost saving strategies that can save a lot of money when deploying applications and infrastructure on the cloud. For example, preemptible instances is a unique feature that can cut the costs of running a VM by 80 percent! A preemptible instance is a normal VM except that Google can decide it needs the capacity back and delete the VM without asking. This can be amazing if you design your applications to be fault tolerant and able to lose a node without disruption. A common use case is having a fleet of preemptible VMs all working on similar tasks. If one is deleted mid-task, the work is shifted to another VM.

Other cost saving features using GCP are per-minute billing, committed use discounts, sustained use discounts, recommendations engine, etc.

Hint: Cloud architects will want to be aware of these features to optimize costs within GCP for their organization.

Networking

Google has completely redesigned network infrastructure from the ground level in order to accommodate their unparalleled scale. They have a global, private, high-speed fiber network with custom routing protocols, hardware and even topology. GCP sits on this networking stack but completely resolves you of managing the complexity of a physical network. This totally changes how architects and developers need to think about networking and greatly simplifies management of firewall rules and routing, for example.

Network objects in GCP are global. This means you could have a VM in Hong Kong and a VM in Belgium on the same private network that are still able to communicate. You could even toss a global load-balancer in front of them and have your customers near both locations refer to your services by the same IP. These networks are also not tied to a single IP space, meaning you can have completely unrelated subnets (such as 10.1.1.0/24 and 192.133.71.0/24) in the same private network able to communicate. This communication between subnets can be easily filtered via firewall rules.

Firewall rules can be implemented two different ways within GCP. Traditional "iptables style" rules that specify which IP ranges can communicate with other IP ranges over certain protocols can be created with specified priority values to determine which rule to apply in a given scenario. This can be complex since you need to know which servers are on certain IP ranges because if they change, you must update the rules. GCP allows for configuring firewall rules that are just as powerful via network tags. This means a network architect can control traffic across their network by simply tagging resources.

Example rules:

Name	Targets	Source Filters	Protocols / Ports	Action	Priority	Network
database-traffic	backend	Tags: frontend	tcp:3306	Allow	1000	default
public-web-traffic	frontend	IP ranges: 0.0.0.0/0	tcp:80	Allow	1000	default

This results in traffic being allowed from the internet only to VMs with the frontend tag and allows the frontend and backend VMs to communicate with each other.

Storage

Applications are typically not very useful unless they have access to your data. There are numerous storage options hosted on GCP including persistent disks, cloud storage and database solutions.

Persistent Disks

This is block storage for your VMs. Standard HDD and SSD options are available to attach to your VMs. These live independently of the VM and can be attached to multiple VMs concurrently. Google automatically handles data redundancy and performance scaling behind the scenes. Local SSDs, which are physically co-located on the host of the VM, are available for high performance applications.

Cloud Storage

Cloud storage is Google's answer for an object store. Arbitrary blobs of data are uploaded into a "bucket" and then can be versioned, widely-replicated and shared. Cloud storage has multiple storage classes: regional, multi-regional, nearline and coldline. Multi-regional storage buckets are automatically geo-redundant, which means all data is replicated across multiple data centers and leaves your data safe if a whole data center goes offline.

Nearline and coldline storage buckets are just as performant as the other storage classes for retrieving data but allow for balancing the costs of retrieval versus storage. For example, coldline is the cheapest storage option but has the highest cost of retrieving data. This is ideal for backing up large amounts of data that may never need to be accessed.

Database Solutions

Google has two NoSQL solutions: Datastore and Bigtable. For relational data, Google has managed Cloud SQL instances (MySQL or Postgres) and Cloud Spanner. Cloud Spanner is the world's first relational database that offers the ability to scale to thousands of servers while maintaining high performance and strong consistency.

In this next part, we will discuss how these core infrastructure pieces can be augmented, so we can seamlessly scale to massive proportions with no intervention by a systems administrator—all while maintaining the lowest costs possible given the workload.

Scaling and High-Availability

Most enterprise level applications have a need for multiple copies of a service to run concurrently. This may be to handle more workload by simply having more instances of the service (horizontal scaling) as opposed to vertical scaling by increasing the vCPUs or RAM of the server. Having multiple copies of a service in separate locations is valuable in case one copy crashes or an accident destroys a physical host. To scale in a highly available manner, where the end user will not experience an interruption if an instance goes down, requires a single endpoint that redirects user traffic to healthy instances. GCP provides tools to automatically scale the number of instances of a service and to direct traffic across them.

Instance Groups

Instance groups allow you to bundle your VMs together for load-balancing and manageability purposes. They can also be configured with an instance template that allows the instance group to automatically create additional instances if demand increases (auto-scaling) or an existing VM crashes (auto-healing).

Load Balancers

If a natural disaster disrupts a data center or a hardware failure fries a rack of servers, a load balancer can detect any unhealthy or absent VMs and direct customer traffic to a healthy instance without intervention. Since GCP networks operate at a global scale, this could mean the users that are normally directed to the Sydney data center are temporarily directed to Singapore instead. GCP makes this automatic via anycast IPs that route a user to the nearest VM which can satisfy the request. This means a cloud architect no longer needs to design a routing solution to handle users from different regions. For example, instead of having users in Australia visit www.example.com.au, there can be a single domain such as www.example.com that all users in the world can use.

Auto-Scaling

An instance group can be set to automatically scale based on any metric, such as CPU usage or number of connections. This is enabled simply by checking a box stating you want auto-scaling and providing the target values for certain metrics. This alleviates the need for a systems administrator to wake up to a late-night page in order to provision another machine to handle an unexpected increase in workload.

The auto-scaler will also delete VMs when workload decreases. This could be a huge savings for applications with long lulls in workload as you will not pay for VMs to sit idle. A common case is a website targeted to a particular region, say a state government website, where most traffic will be during waking hours for that given region.

Automated Infrastructure

Every Google product has a REST API allowing for automating provisioning, maintenance, and monitoring tasks. These APIs can be explored via the APIs Explorer.

Deployment Manager

Deployment Manager is a hosted tool that allows you to define the entire infrastructure needed by your application in template files. This allows you to version control your infrastructure definition. More importantly, it enables exact clones of your infrastructure to be deployed multiple times. There are numerous other benefits to defining your Infrastructure-as-Code.

Perhaps you would like multiple environments, such as Development, Staging, and Production, for your application in order to promote the latest versions of your application code. Deployment Manager allows you to define the infrastructure once but deploy to each of these environments. Your workflow could be to promote an application version from one environment to the next.

First, deploy both the infrastructure and application to Development whenever there is a new version. Then, when that version has been tested and blessed, deploy the same version of the infrastructure and application to Staging. Finally, deploy the version to Production. It is likely that any infrastructure misconfigurations will be identified in the earlier environments as each environment will be deploying near identical infrastructure.

The practice of defining your infrastructure in this way has many other benefits as well. Instead of creating complicated change management requests when a new piece of infrastructure is required or configuration needs to change; developers, administrators, or operators can make the change themselves to the code repository that defines the infrastructure. Then these changes could be peer-reviewed, merged, and then the infrastructure is updated automatically. This also promotes the coordination between development and operations (DevOps) by removing barriers between the teams and processes.

Cloud Launcher

Cloud Launcher builds on Deployment Manager by allowing various third parties to upload their infrastructure definitions to a kind of market place. For example, you can spin up an entire WordPress site by clicking a button in Cloud Launcher. This will provision the required VMs and storage and then configure the software—all of which is defined in a Deployment Manager template for you.

Day 2 of the “Architecting with Google Cloud Platform: Infrastructure” course discusses these infrastructure augmentation tools and others in-depth. Additional topics covered include automatically bootstrapping VMs via start-up and shut-down scripts, monitoring your cloud resources, and utilizing the GCP Cloud SDK for complete control over GCP resources.

While deploying to the cloud removes the burden of managing hardware, these automation tools further simplify the initial and ongoing management of infrastructure. Google takes this yet another step forward by providing a number of Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS) solutions that can ease the responsibility of the systems administrator and give more power to the developer.

We've now addressed infrastructure services and augmented infrastructure. In the final sections of this paper, we will discuss the fully managed services Google provides, which allows you to get work done without ever being concerned with infrastructure with GCP's Platform-as-a-Service (PaaS) solutions.

PaaS solutions were some of the first services offered by cloud providers. These solutions came early since they provide a tremendous value of allowing a developer to deploy their code without having to request or provision infrastructure, which may take days or weeks. GCP has a number of such PaaS solutions that allow you to deploy your application without the complexity of provisioning and managing infrastructure. These solutions fall into two families: serverless applications and Linux containers.

Serverless Apps

Maybe you are a developer that has some code that you just want to run and not worry about infrastructure at all. Google App Engine (GAE) and Cloud Functions allow you to upload your code to Google and have it run when needed.

GAE will deploy your application for you and scale to near infinity if needed without any intervention. This is ideal for web applications or mobile application servers as you may not know the scaling demands ahead of time such as traffic spikes from online shopping during Black Friday or if a video hosted on your site goes viral. GAE also allows for performing rolling updates of your application or splitting traffic across two versions of the application for A/B testing. GAE should be considered as a platform to use when deploying lightweight applications that are agnostic to the infrastructure.

Cloud Functions, on the other hand, take the classic cloud use case of "pay-as-you-go" to the next level. You can have your code only run when you need it and be billed at the precision of the nearest 100 milliseconds of compute time. This means you only pay for the compute you use instead of any idle time waiting for a server to start or the next request to come in. There are various use cases for running a simple code function on demand. Mobile applications often communicate with a backend server to perform actions such as updating user information, saving a game, or sending an email. All of these cases can be solved by triggering the corresponding cloud function when a user performs the action in the mobile app.

Containers

Linux containers are the current buzzword in the tech industry. Their appeal is that they are much lighter weight than VMs and provide complete isolation of a running process. Docker images are a way to define how your application and its dependencies are bundled together. Docker has the benefit of being an industry standard, which allows you to use the same definition inside and outside of GCP. There are at least three ways to run standard Docker containers on GCP as of the time of publication: Google App Engine, VMs on Google Compute Engine (GCE), and Google Container Engine (Kubernetes).

Google App Engine (GAE) Flex

This is the simplest way to deploy a Docker container on GCP. Just like the traditional GAE described, your service could scale and deploy automatically. The only difference is you must provide a Dockerfile describing how to create the container image from your application. From there, GAE will do the rest.

Deploying VMs with Docker Images (Alpha Feature)

This mechanism works a lot like GAE Flex does under the covers. A standard Docker image can be specified when creating a VM. Then, a VM will be provisioned with Docker installed and the specified container running. This allows you to deploy containers but get all the manageability of GCE such as load balancers, networks, and cost saving strategies.

Google Container Engine (GKE)

Kubernetes is an orchestration engine for deploying containers across many hosts and allows you to scale and spread your applications. Deploying a Kubernetes cluster typically requires provisioning multiple nodes, installing Kubernetes, and configuring the cluster. GKE allows you to specify how many nodes you would like and handles the rest for you.

If you already have a Kubernetes cluster on-premises, you can burst to the cloud by setting up federation between your on-prem cluster and a cluster in GKE. This allows you to distribute your applications to multiple cloud providers or reach a scale that your on-prem infrastructure cannot support.

Conclusion

Throughout this paper, “Understanding Google Cloud Platform: Architecture,” we discussed working at three different levels of the technology stack.

First, we focused on deploying infrastructure on which to provision the system architecture. Then, we built on infrastructure, discussing tools that improve management of large-scale systems and further automate system operations. And finally, we moved past infrastructure and discussed tools available that all allow deployment of applications directly.

Outside of this white paper, there is a host of Google product families. GCP also provides tools for Identity and Access Management, Machine Learning, Big Data, Developer Tools, etc. Many of these products are discussed in depth in various GCP courses offered by Global Knowledge, such as the Google Cloud Platform Fundamentals: Core Infrastructure for an overview of all the products and services, "Data Engineering on Google Cloud Platform" training course for all things Big Data and Machine Learning or the "GCP Fundamentals for AWS Professionals" training course, which introduces the seasoned AWS pro to Google Cloud.

Learn More

As one of a select few worldwide Google Cloud Premier Training Partners, our authorized training courses enable users to demonstrate proficiency using Google Cloud Platform (GCP) products and services. Gain the necessary skills to develop, manage and administer solutions or design and build data processing systems for big data and machine learning.

Learn how to incorporate GCP into the business strategy and advance to a technical deep-dive on different solutions in a hands-on environment taught by Google certified instructors. View our Google Cloud Learning Path to see how Global Knowledge can help you and your team progress your skills. And, if you're ready to validate your expertise, check out the Google Cloud certifications.

Visit www.globalknowledge.com or call 1-800-COURSES (1-800-268-7737) to speak with a Global Knowledge training advisor.

About the Author

Alex Meade is an instructor with Global Knowledge as well as a DevOps consultant. He was a core contributor for five years early in the popular, open-source OpenStack project and thus has a strong understanding of "the cloud" and large, complex systems. Alex's recent work has been focused on DevOps, Docker containers, and the Google Cloud Platform. He currently does all of this while traveling and living all over the world.