# Getting To Know Google Compute Engine

## And How To Use It

BY DAVID POSIN

CONTENTS



## WHAT IS THE GOOGLE COMPUTE ENGINE?

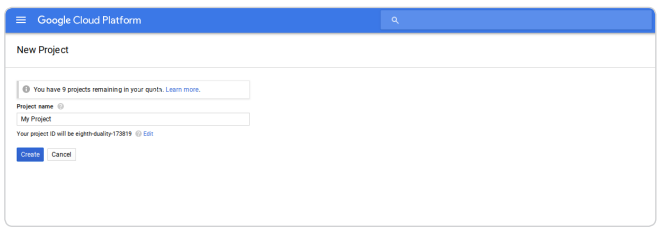
The Google Cloud Platform is a suite of product offerings designed to bring the robust flexibility of their cloud architecture to their customers. It is similar in nature and function to Amazon's AWS and Microsoft's Azure. Google entered the cloud computing marketplace later than its competitors. The late entry may have been an initial stumbling block, but that is no longer the case. Google has reached feature parity with its competitors. Google's virtual machine offering, called the Google Compute Engine, is every bit as robust as an Amazon EC2 or Microsoft Azure instance.

Google Compute Engine is Google's Infrastructure-as-a-Service (IaaS) virtual machine offering. An IAAS platform replaces, or supplements, traditionally onsite network infrastructure assets, like servers and routers, with cloud-based products that perform the same functions. The Compute Engine allows customers to use powerful virtual machines in the cloud as server resources instead of acquiring and managing server hardware.

Customers can configure and run a wide variety of virtual machine configurations. Google provides Linux and Windows as operating systems for their machines, although a custom machine option means being able to run any OS on any image you maintain. The virtual, and in some cases physical, hardware added to the machine helps to dictate its purpose. For example, a web application server might need lots of RAM and CPUs (Central Processing Units), but it does not need GPUs (Graphics Processing Units). Alternatively, a server working on modeling components or streaming media can use configured GPUs. A simple website server might not need any of this, instead opting for a medium amount of RAM and CPU power.

The Google Compute Engine feature set can meet the virtualization requirements of any enterprise. This card serves as an introduction to the Compute Engine and explores its main features. The card also acts as an easy portal to many of the documentation pages you may need when setting up a VM instance.

## WHAT IS A PROJECT?




A project is the main organizing unit for instances. The first step in using Google's Compute Engine is to make a project. All instances and resources are then created in that project. Instances and resources in a project are unique to that project, and cannot be managed by resources in other projects. Resources from other projects can still communicate with each other over standard network communication protocols but they can not be managed together.

## VIRTUAL MACHINE FEATURES

### MACHINE TYPES

The Machine Type describes the virtual hardware attached to an instance including RAM and CPUs. It also sets potential limitations such as the maximum number of persistent disks, GPUs, and disk space allowed. There are two main types of machines: Predefined and Custom.

#### PREDEFINED MACHINE TYPES

Predefined machine types are pre-configured virtual machine templates that you can use to set up your virtual machine. The configurations have been pre-optimized by Google and meet most needs. Google has broken the predefined machine types into four categories that range in purpose:

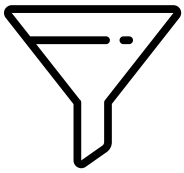
| CATEGORY | PURPOSE |
|---|---|
| Standard | Balanced between processing power and memory. Fits most common application needs |

RAPID7

# The Modern Solution to Infrastructure Monitoring and Troubleshooting

No complex setup. No waiting. Just answers.

Rapid7 InsightOps combines log management with IT asset search for a new approach to infrastructure and application monitoring and troubleshooting. It takes only minutes to setup and is free to try.



## Centralize

Easily centralize data across your infrastructure, from system logs to IT assets.

## Monitor

Monitor your IT environment for anomalies, key metrics, and critical events.

## Answer

Get the answers you need, when you need them, to resolve critical issues and maintain uptime.

**Start a Free Trial**

| CATEGORY | PURPOSE |
|---|---|
| High-Memory | Emphasis is put on memory over processing power for tasks that need accessible non-disk storage quickly |
| High-CPU | Higher CPU usage for high-intensity applications that require processing over memory |
| Shared-core | A single virtual CPU, backed by a physical CPU, that can run for a period of time. These machines are not for use cases that require an ongoing server or significant power. The micro shared-core machine also provides bursting capability when the virtual CPU requires more power than the single physical core. Bursting is for a short, intermittent period based on need. |

### CUSTOM MACHINE TYPES

It is possible to fully configure the virtual hardware manually for a Compute Engine VM instance. Administrators can select the number of virtual CPUs and memory allocated to a VM instance within the boundaries set by Google. A certain amount of memory is required based on the number of virtual CPUs, and vice versa. Only use a custom machine type for specific purposes not meet by the predefined types.

### GPUS

A GPU is for intensive operations that require dedicated processing units or graphics processing. This includes activities such as computer-aided design, visual modeling, data modeling, machine learning, and streaming. The feature to add GPUs to a Compute Engine is in beta as of June 2017.

**Quick Tip**: Instances with loaded GPUs will be forcefully terminated during maintenance events by Google. This is not true of instances without GPUs which are managed automatically without downtime by Google. To know when a GPU loaded instance is going to be terminated, monitor the `/computeMetadata/v1/instance/maintenance-event` endpoint from the instance. If the return value is a timestamp, then the returned time is when the machine will be terminated.

## STORAGE

### PERSISTENT DISKS

Persistent disks are the simplest, and probably most commonly used, type of storage for a standard Google Compute virtual machine instance. They mimic the feel and functionality of any standard disk drive that you might attach to a server. In reality, persistent disk resources stretch your data across multiple volumes to ensure reliability and redundancy.

Persistent disks need to be mounted onto the machine once created. Once mounted, interaction with a persistent disk feels like working with a normal volume. The Compute Engine stores the disk data independently of the instance. A persistent disk's life cycle is not tied to a particular instance, making them flexible and easily migrated. Persistent disks are also encrypted, and data is encrypted the moment it leaves the instance. It is possible to override some of Google's automated optimizations of a persistent disk so that you can manage its configuration directly, but that is recommended only for specific purposes by experienced administrators.

There are two categories of persistent disks:

- **Shared**: Standard type of disk storage with average read/write speeds.
- **SSD**: (Solid State Drive) Higher read/write speeds for instances that require improved performance.

**Quick Tip**: Linux operating systems are also capable of creating near zero latency file systems in memory called RAM disks. You can mount a RAM disk to your virtual instance for caching and other application purposes. Mount a disk using the tmpfs disk type to create a RAM disk. For example: `sudo mount -t tmpfs -o size=25g tmpfs /mnt/ram-disk`. A RAM disk will not be backed up like a normal persistent disk, so saving anything stored there must be done manually.

### LOCAL SSD

A Local SSD is physically attached to the virtual machine running it. The SSD greatly improves performance and reduces latency. However, the improved performance comes with a trade-off: the data on a local SSD only persists while the instance is running. Stopping an instance clears the data from any SSD that is locally attached. The close connection also makes this the most expensive disk option.

These drives are good for cache data that is not stored for long term use or redundant data that can be easily rebuilt. Local SSDs cannot stretch across zones or be load balanced. Data is still encrypted.

### CLOUD STORAGE BUCKETS

The least expensive storage options are Cloud Storage Buckets. There are several classes of Storage Buckets depending on the scope and level of performance needed. Some storage classes of Buckets can reach performance levels similar to a persistent disk, but their performance is less consistent.

Buckets can exist globally and can be touched by multiple instances. A persistent disk is limited to a Zone, making the scope of a Cloud Storage Bucket the widest ranging of all storage options. There is an added complexity to its easy accessibility, since instances can potentially overwrite each other's data. The more global a class is in scope, the slower and more latent its operation. In practice, the more global in scope, the more likely it is used for long-term archival and other non-mission-critical tasks.

Cloud Storage Buckets are automatically redundant, making them extremely reliable. The Google Compute Engine encrypts all data just like the other storage options. Buckets are mountable onto a VM's file system just like a persistent disk.

The main difference between a bucket and a persistent disk is the nature of the data stored. Persistent disks are file-based storage mechanisms, while Cloud Storage Buckets are object stores. Since they are object stores, they cannot serve as a root drive for a VM.

## IMAGES

Images are the software applied to your instance, including its root operating system.

### PUBLIC IMAGES

Google provides a set of public images. These images are a collection of open source options and proprietary options. The proprietary versions exist in a premium tier and incur extra costs. Public images serve as a starting point for most virtual machine instances and come packaged with only the operating system. The list of public images, as of June 2017, is:

| IMAGE OS | PREMIUM |
|---|---|
| CentOS | |
| Google Container-Optimized OS | |
| CoreOS | |
| Debian | |
| Red Hat Enterprise Linux | Yes |
| SUSE Enterprise Linux Server | Yes |
| SLES for SAP | Yes |
| Ubuntu | |
| Windows Server | Yes |

Google also offers an additional flavor of Windows Server with SQL Server pre-installed:

| SQL SERVER VERSION |
|---|
| SQL Server Enterprise |
| SQL Server Standard |
| SQL Server Web |
| SQL Server Express |

### CUSTOM IMAGES

It is possible to use an image pre-loaded with software that more closely meets your needs. The public images are a good starting point, but they are designed to be built upon and turned into custom images.

Production ready environments are built on custom images. A custom image should not only have the software needed; it should have all the scripts necessary for the instance to work automatically without administrator intervention. Google can start and stop custom images for load balancing or recovery purposes.

**Quick-Tip**: Metadata about an instance is stored at the http://metadata.google.internal/computeMetadata/v1/ endpoint. Images are spun up into instances dynamically, so anything that is used at runtime, such as an automation script, will need to retrieve instance attributes through this endpoint. For more information, see Storing and Retrieving Instance Metadata.

### IMAGE FAMILIES

Image families are a way to organize images so that the desired version is available without updating images and code every time the version is changed. Using an image family is similar to providing an alias so that installation can use a non-version numbered name and still get the most recent stable version. Images are added to, deleted from, and deprecated in an image family. When calling that image family, it returns the most recent non-deprecated version. Rolling image versions forward or backward in an image family is as simple as adding a new version or deprecating an existing version.

## INSTANCE GROUPS

Managing large numbers of individual virtual machines can be cumbersome. Google helps administrators manage this potential workload with the use of instance groups. Administrators can manage the instances in an instance group simultaneously. There are two types of instance groups: managed and unmanaged.

### MANAGED

A managed instance group is made up of instances all built from the same image template. An image template defines all the attributes for an instance, including which image to use. Administrators and the Compute Engine can manage the instances in bulk since all the instances are the same.

This group is the most common, and the one Google suggests administrators use. A managed instance group can automatically scale and be used to balance server loads. Since the instances are identical, the Compute Engine can bring new instances up or down to match the current traffic. The Compute Engine can also stop unhealthy instances and replace them with new instances.

### UNMANAGED

An unmanaged instance group is a collection of different instances, not based on the same image template. Management options are limited for an unmanaged instance group since the instances are not identical. An unmanaged instance group cannot be used for load balancing and cannot auto scale.

### PREEMPTIBLE VM INSTANCES

Preemptible instances are a low-cost option for non-mission critical uses. They run when the Google Compute Engine has resources available. Alternatively, the Google Compute Engine will terminate a preemptible instance when its resources are needed. Preemptible instances are perfect for uses that can function with variable processing power and support variable processing times, such as batch operations and data archival. Any task being performed by a preemptible instance should be fault-tolerant.

Preemptible instances can be part of an instance group. The Compute Engine can terminate an instance in a group when it needs resources, but is highly unlikely to terminate all the instances in a group. Removing instances from the group causes the service to slow but not stop entirely. When resources are again available, the Compute Engine attempts to restore the instance group to its quota.

There are a few things to keep in mind when using a preemptible instance:

- There is no guarantee about the amount of power available at any one time.
- Instances are shut down in the order of most recent to longest running. Shutting down the newest first is unlikely

to jeopardize a long running process that may have made progress in favor of a process that may have just started.

- Preemptible instances can run a maximum of 24 hours before being terminated by the Compute Engine system.
- They are not covered by any SLAs since they are variable by their very nature.
- It is best to plan for system enforced termination with scripts that can run to prepare the process for exiting. The instance is notified 30 seconds before termination for shutdown scripts to run.

## REGIONS AND ZONES

Google Cloud resources exist in one of the three location categories:

- Region: a broad geographical area. Regions are made up of zones. Resources at the regional level are called Regional resources.
- Zone: a unit of resources that make up regions. Resources at the zone level are Zonal Resources.
- Global: A resource that exists in no specific region and can be used in any region. These are labeled Global resources.

The scope of a resource is constrained to its location category. For example, a zonal resource like a persistent disk can only be used by an instance in the same zone. Meanwhile, regional resources are accessible to other regional resources and zonal resources in that region. Resources, like images, exist globally and can be used by a resource in any region.

As of June 2017, Google had the following Regions and Zones:

| REGION | ZONES |
|---|---|
| Americas | us-central1 |
| | us-east1 |
| | us-east4 |
| | us-west1 |
| Europe | europe-west1 |
| | europe-west2 |
| Asia | asia-east1 |
| | asia-northeast1 |
| | asia-southeast1 |
| Australia | australia-southeast1 |

## ACCESS CONTROL

- The only user automatically able to use a project and its resources is the project creator. Users, servers, and external integrations are connected to the project resources manually.
- Users: user access is controlled through project roles, like "Compute Engine Network Admin," or primitive roles like "Editor."
- Service Accounts: Applications and resources can be given access to each other through service accounts. Service accounts are specific to resources and relieve developers from using user credentials for systems to talk.
- SSH Access: For users to access a VM through SSH but to have no project level access, add a user's public key to the project or a specific instance.

## INTERACTING WITH THE COMPUTE ENGINE

There are several ways to interact with Compute Engine instances: the gcloud CLI, a REST API, and the console. The console is a graphical interface that is straightforward to use, although encumbered by UI operations. The gcloud CLI tool is accessible on all Compute Engine VMs when started through the console's ssh option. The gcloud tools can also be installed onto any Compute Engine VM if you need to connect directly through SSH or some other means. Finally, there is a REST API that can be used to manage Compute Engine VMs.

### COMPUTE GROUPS

The operations that an administrator performs on a Compute Engine VM is broken into categories, called Groups. The Groups, their descriptions, and links to their documentation are all provided below:

**Quick Tip**: Commands for using these groups on the command line will be `google compute [compute group]`

**Quick Tip**: To explore/interact with the API endpoints without writing code visit the Oauth 2.0 Playground

| GROUP | DESCRIPTION | GCLOUD | API |
|---|---|---|---|
| **Accelerator Types** | Type of acceleration used by the GPUs that can be attached to a VM | accelerator-types | Accelerator Types |
| **Addresses** | Create, release, and list the addresses associated with the current VM | addresses | Addresses |
| **Autoscalers** | Resource to allow for a group of instances, called an instance group, to scale based on demand automatically | | Autoscalers |
| **Backend Buckets** | Manage Google Cloud Storage buckets connected to the current VM. Backend buckets are multiregional highly-available storage areas. They are an excellent place to store static resources for a website or application. | backend-buckets | BackendBuckets |
| **Backend Services** | Configures and manages a backend service. A backend service load balances a group of backends, which are instances inside a group of instances, called an instance group, that can all perform the same task. A backend service knows which instance has how much traffic, which region it is in, and can maintain sessions across backends. | backend-services | BackendServices |

| GROUP | DESCRIPTION | GCLOUD | API |
|---|---|---|---|
| **config-ssh** | Automatically adds an entry for each instance to the user's ssh file | config-ssh | |
| **connect-to-serial-port** | Allows users to connect to another VM over ssh to the VM's serial port | connect-to-serial-port | |
| **copy-files** | DEPRECATED. Use scp instead | DEPRECATED | |
| **Disk Types** | Retrieve the disk type information about disks in the project | disk-types | DiskTypes |
| **Disks** | Manage the disks in a project including adding, removing, resizing, and taking snapshots | disks | Disks |
| **Firewalls** | Manage the firewall rules on a Compute Engine VM | firewall-rules | Firewalls |
| **Forwarding Rules** | Manage the forwarding of traffic to a pool of VMs or backend services | forwarding-rules | ForwardingRules |
| **Global Addresses** | Used for configuring Global Forwarding Rules used for http load balancing | | GlobalAddresses |
| **Global Forwarding Rules** | Forwards traffic to the load balancers for HTTP load balancing | | Global ForwardingRules |
| **Global Operations** | Operations that have global implications, and are therefore implemented at a global level. For example, adding a new image. Since images can be used by any instance in any zone, inserting a new image is a global operation | operations | Global Operations |
| **Health Checks** | To check the status of virtual machines being used by a load balancer. Allows for checking over non-http connections | health-checks | HealthChecks |
| **HTTP Health Checks** | Manage the health status of load balancer instances, specifically over http | http-health-checks | HttpHealthChecks |
| **HTTPS Health Checks** | Manage the health status of load balancer instances, specifically over https | https-health-checks | HttpsHealthChecks |
| **Images** | Manipulate the global library of project vm images | images | Images |
| **Instance Group Managers** | Manage the instance group entity | | InstanceGroup-Managers |
| **Instance Groups** | Groups of instances bundled into identifiable groupings for easier management | instance-groups | InstanceGroups |

| GROUP | DESCRIPTION | GCLOUD | API |
|---|---|---|---|
| **Instance Templates** | Configuration settings for use in launching and deploying new instances | instance-templates | Instance Templates |
| **Instances** | Manage an existing Compute Engine instance | instances | Instances |
| **Licenses** | Read-only. Retrieve informantion on the software licenses used throughout Google Compute resources | | Licenses |
| **Machine Types** | Read-only. Information on the machine types used in existing instances | machine-types | MachineTypes |
| **Networks** | Manage, and peer on, the networks included in a project | networks | Networks |
| **Project Information** | Mostly read, but some very constrained configuration management, of projects. Projects are designed to mostly be managed through the console. | project-info | Projects |
| **Region Autoscalers** | Manipulate autoscaling policies for instances in managed instance groups on a regional level | | Region Autoscalers |
| **Region Backend Services** | Manipulate a group of virtual machines functioning backend services on regional level | | RegionBackend Services |
| **Region Instance Group Managers** | Manage an instance group | | RegionInstance-GroupManagers |
| **Region Instance Groups** | Retrieve information about an instance group | | RegionInstance-Groups |
| **Region Operations** | Get or delete an operation performed on a regular resource, such as an IP address update | | Region Operations |
| **Regions** | List Compute Engine regions, or get information on a specific region | regions | Regions |
| **Reset Windows Password** | Reset a user's Windows password, or create a user on a windows virtual machine. Use with care as it can result in lost encrypted data if not done correctly | reset-windows-password | |
| **Routers** | Administrate router resources | routers | Routers |
| **Routes** | Control the routing tables for Google Compute virtual machines | routes | Routes |
| **scp** | Copy files to and from a Compute Engine virtual machine. Transfer occurs over scp or pscp (Windows) | scp | |

| GROUP | DESCRIPTION | GCLOUD | API |
|---|---|---|---|
| **Snapshots** | Manage existing persistent disk snapshots, including deleting and adding labels | snapshots | Snapshots |
| **ssh** | Use ssh to interact with another Compute Engine virtual machime | ssh | |
| **SSL Certificates** | Administer ssl certificates on a virtual machine, including creation and deletion | ssl-certificates | SslCertificates |
| **Subnetworks** | Modify, or get information on, subnetworks in a project's network | | Subnetworks |
| **Target HTTP Proxies** | Manage a network's target http proxies. Target Http Proxies are used by global forwarding rules to route traffic to url maps. | target-http-proxies | Target HttpProxies |
| **Target HTTPS Proxies** | Manage a network's target https proxies. Target Https Proxies are used by global forwarding rules to route traffic to url maps. | target-https-proxies | Target HttpsProxies |
| **Target Instances** | Create, delete, and get information on instances designed to be terminating endpoints | target-instances | TargetInstances |
| **Target Pools** | Manipulate a project's target pools. Target pools are a collection of instances, their associated health checks, and fallback target pools | target-pools | TargetPools |
| **Target SSL Proxies** | Create, delete, and update network ssl proxies, which provide authentication and encryption to a backend service | target-ssl-proxies | TargetSslProxies |
| **Target YCP Proxies** | Manage tcp proxies for passing requests through the network to backend services | target-tcp-proxies | TargetTcp Proxies |
| **Target YPN Gateways** | Manipulate Compute Engine VPN gateways | target-vpn-gateways | TargetVpn Gateways |
| **URL Maps** | Modify the mapping between URL and backend service | url-maps | UrlMaps |
| **Vpn Tunnels** | Manipulate Compute Engine VPN tunnels | vpn-tunnels | VpnTunnels |
| **XPN** | Allow for the sharing of Google Compute's VPC (Virtual Private Cloud) across projects | xpn | |
| **Zone Operations** | Resources representing per-zone operations, such as inserting an instance into a zone | | ZoneOperations |
| **Zones** | Get information about Google Compute zones | zones | Zones |

## ABOUT THE AUTHOR

**DAVID POSIN** has been involved in the Information Technology Industry for two decades. Fifteen years of that time was spent consulting with many companies in a wide range of industries to build solid technology stacks and robust application architectures. David has watched the Cloud and the World Wide Web grow from their infancy, and now spends every day fully entrenched in those worlds. Currently, David builds high-performance web applications and offers professional technical writing services.



DZONE, INC.
150 PRESTON EXECUTIVE DR.
CARY, NC 27513

888.678.0399
919.678.0300

REFCARDZ FEEDBACK WELCOME
refcardz@dzone.com

SPONSORSHIP OPPORTUNITIES
sales@dzone.com