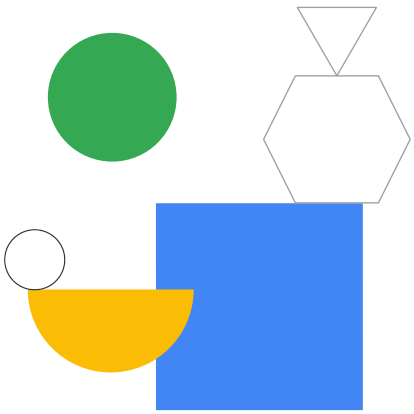
Google Cloud

Virtual Machines



In this module, we cover virtual machine instances, or VMs.

VMs are the most common infrastructure component, and in GCP, they're provided by Compute Engine. A VM is similar, but not identical, to a hardware computer. VMs consists of a virtual CPU, some amount of memory, disk storage, and an IP address. Compute Engine is GCP's service to create VMs; it is very flexible and offers many options, including some that can't exist in physical hardware. For example, a micro VM shares a CPU with other virtual machines, so you can get a VM with less capacity at a lower cost. Another example of a function that can't exist in hardware is that some VMs offer burst capability, meaning that the virtual CPU will run above its rated capacity for a brief period, using the available shared physical CPU. The main VM options are CPUs, memory, disks, and networking.

Agenda

01	Compute Engine Lab: Creating Virtual Machines
02	Compute Options (vCPU and Memory)
03	Images
04	Disk Options
05	Common Compute Engine Actions Lab: Working with Virtual Machines

Now this is going to be a very robust module; there's a lot of detail to cover here with how virtual machines work on GCP. First we'll start with the basics of Compute Engine, followed by a quick little lab to get you more familiar with creating virtual machines.

Then, we'll look at the different CPU and memory options that enable you to create different configurations.

Next, we will look at images and the different disk options available with Compute Engine.

After that, we will discuss very common Compute Engine actions that you might encounter in your day-to-day job. This will be followed by an in-depth lab that explores many of the features and services covered in this module.

Let's get started with an overview of Compute Engine!

Google Cloud compute and processing options

	Compute Engine	GKE	App Engine standard environment	App Engine flexible environment	Cloud Functions	Cloud Run
Language support	Any	Any	Python Node.js Go Java Ruby PHP	Python Node.js Go Java PHP Ruby .NET Custom runtimes	Python Node.js Go Java .NET Core Ruby Java	Any
Usage model	IaaS	IaaS PaaS	PaaS	PaaS	Microservices architecture	PaaS
Scaling	Server autoscaling	Cluster	Autoscaling managed servers		Serverless	Serverless
Primary use case	General workloads	Container workloads	Scalable web applications Mobile backend applications		Lightweight event actions	Deploy & scale containerized apps

Google Cloud

As mentioned in the introduction to the course, there is a spectrum of different options in Google Cloud for compute and processing. We will focus on the traditional virtual machine instances.

Now the difference is, Compute Engine gives you the utmost in flexibility: run whatever language you want—it's your virtual machine. This is purely an infrastructure as a service or IaaS model.

You have a VM and an operating system, and you can choose how to manage it and how to handle aspects, such as autoscaling, where you'll configure the rules about adding more virtual machines in specific situations. Autoscaling will be covered later in the course.

The primary work case of Compute Engine is any generic workload, especially an enterprise application that was designed to run on a server infrastructure. This makes Compute Engine very portable and easy to run in the cloud.

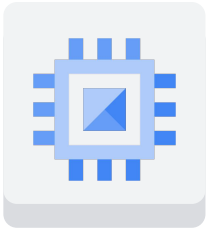
Other services, like Google Kubernetes Engine, which consists of containerized workloads, may not be as easily transferable as what you're used to from on-premises.

Compute Engine

Infrastructure as a Service (IaaS)

Predefined or custom machine types:

- vCPUs (cores) and Memory (RAM)
- Storage
  - Zonal or regional persistent disk (HDD or SSD)
  - Local SSD
  - Cloud Storage
- Networking
- Linux or Windows

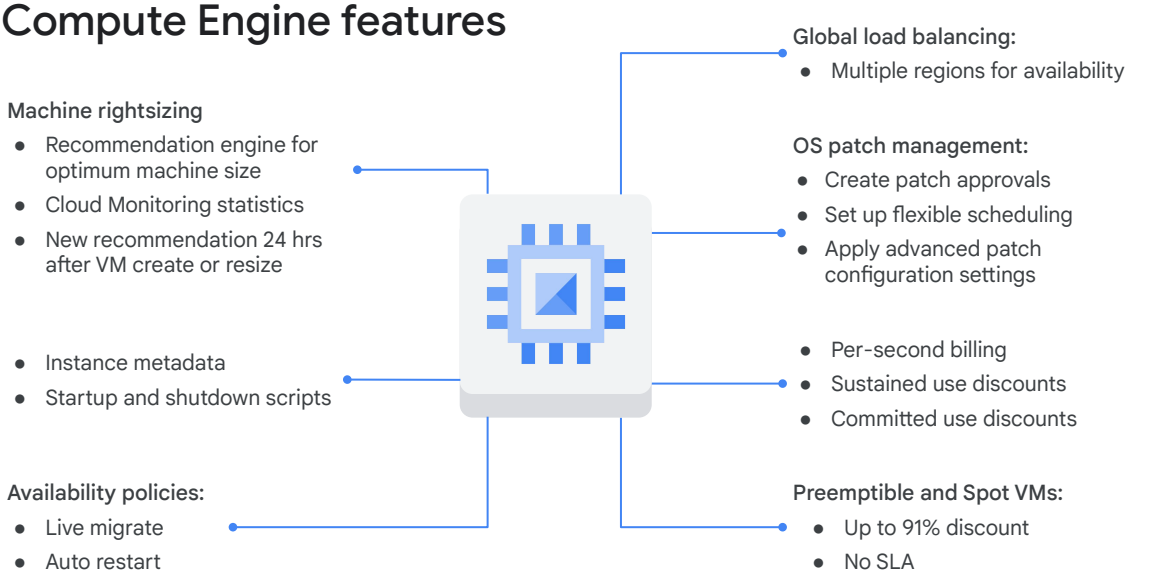


Compute Engine

So what is Compute Engine? At its heart, it's physical servers that you're used to, running inside the Google Cloud environment, with a number of different configurations.

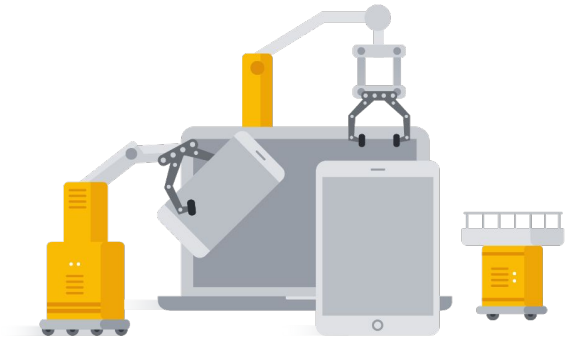
Both predefined and custom machine types allow you to choose how much memory and how much CPU you want. You choose the type of disk you want, whether you want to use persistent disks backed up by standard hard drives or solid-state drives, local SSDs, Cloud Storage, or a mix. You can even configure the networking interfaces and run a combination of Linux and Windows machines. We will discuss these options in more detail later in the module.

Compute Engine features



Several different features will be covered throughout this module, such as machine rightsizing, startup and shutdown scripts, metadata, availability policies, OS patch management, and pricing and usage discounts.

Hardware limitations



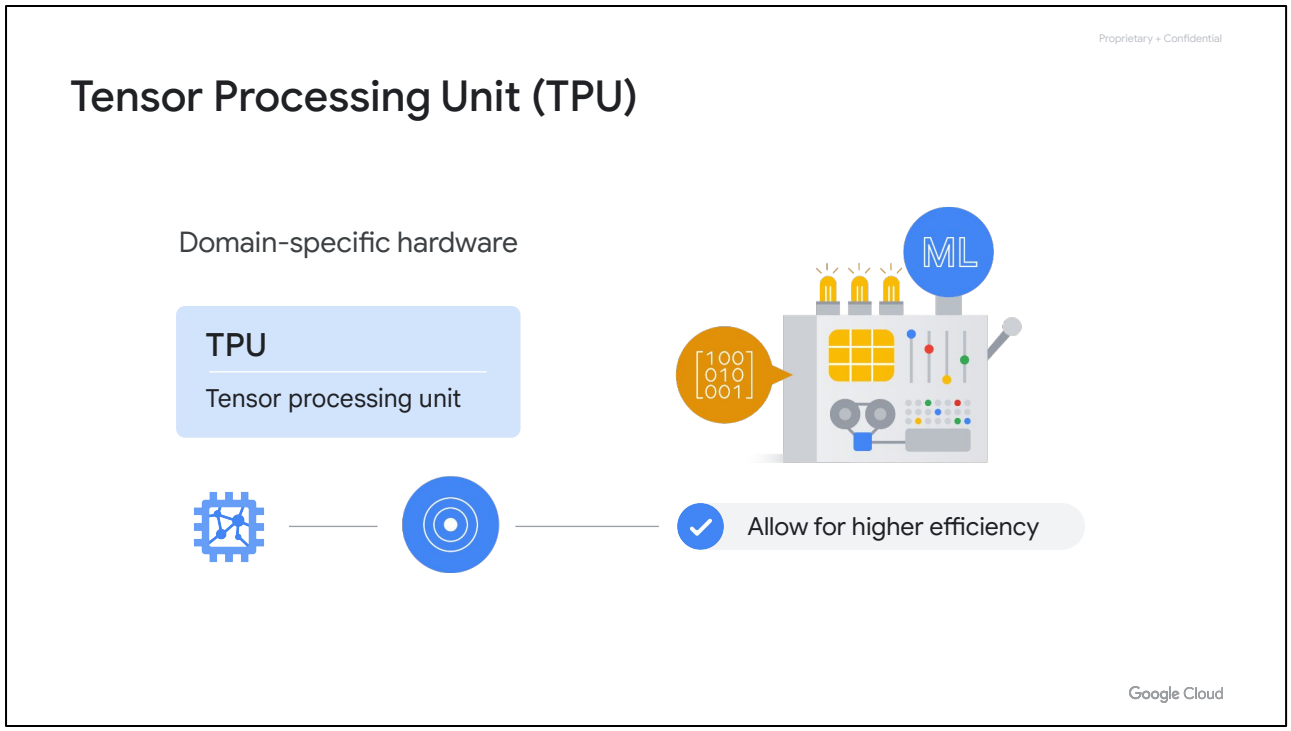
CPU

Central processing unit

GPU

Graphics processing unit

It is important to mention that hardware manufacturers have run up against limitations, and CPUs, which are central processing units, and GPUs, which are graphics processing units, can no longer scale to adequately reach the rapid demand for ML.



To help overcome this challenge, in 2016 Google introduced the **Tensor Processing Unit**, or **TPU**. TPUs are Google's custom-developed **application-specific** integrated circuits (ASICs) used to accelerate machine learning workloads.

TPUs act as **domain-specific** hardware, as opposed to **general-purpose** hardware with CPUs and GPUs. This allows for higher efficiency by tailoring architecture to meet the computation needs in a domain, such as the matrix multiplication in machine learning.

TPUs are generally faster than current GPUs and CPUs for AI applications and machine learning. They are also significantly more energy-efficient. Cloud TPUs have been integrated across Google products, making this state-of-the-art hardware and supercomputing technology available to Google Cloud customers.

TPUs are mostly recommended for models that train for long durations and for large models with large effective batch sizes.

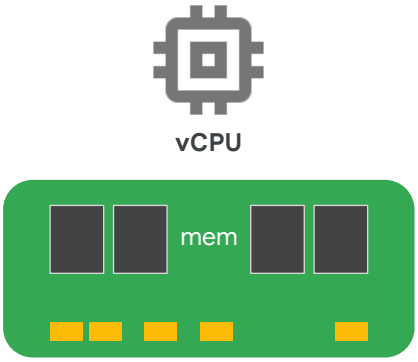
Refer to the documentation for more details.

Compute

Several machine types

- Network throughput scales 2 Gbps per vCPU (small exceptions)
- Theoretical max of 200 Gbps with 176 vCPU

A vCPU is equal to 1 hardware hyper-thread



Let's start by looking at the compute options.

Compute Engine provides several different machine types that we'll discuss later in this module. If those machines don't meet your needs, you can also customize your own machine.

Your choice of CPU will affect your network throughput. Specifically, your network will scale at 2 gigabits per second for each CPU core, except for instances with 2 or 4 CPUs which receive up to 10 gigabits per second of bandwidth.

There is a theoretical maximum throughput of 200 gigabits per second for an instance with 176 vCPU, when you choose an C3 machine series.

When you're migrating from an on-premises setup, you're used to physical cores, which have hyperthreading. On Compute Engine, each virtual CPU (or vCPU) is implemented as a single hardware hyper-thread on one of the available CPU Platforms.

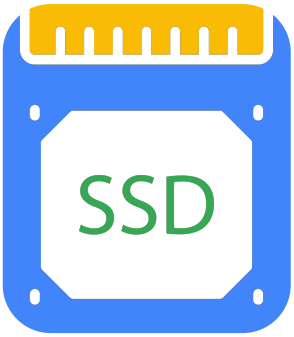
For an up-to-date list of all the available CPU platforms, refer to the CPU platforms documentation https://cloud.google.com/compute/docs/cpu-platforms.

Storage

Disks

- Standard, SSD, or Local SSD
- Standard and SSD PDs scale in performance for each GB of space allocated

Resize disks or migrate instances with no downtime



After you pick your compute options, you want to choose your disk.

You have three options: Standard, SSD, or local SSD. So basically, do you want the standard spinning hard disk drives (HDDs), or flash memory solid-state drives (SSDs)? Both of these options provide the same amount of capacity in terms of disk size when choosing a persistent disk. Therefore, the question really is about performance versus cost, because there's a different pricing structure.

Basically, SSDs are designed to give you a higher number of IOPS per dollar versus standard disks, which will give you a higher amount of capacity for your dollar.

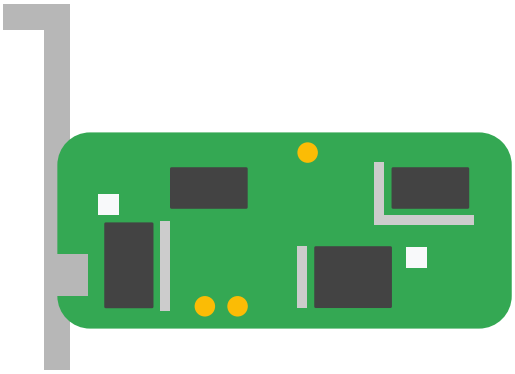
Local SSDs have even higher throughput and lower latency than SSD persistent disks, because they are attached to the physical hardware. However, the data that you store on local SSDs persists only until you stop or delete the instance. Typically, a local SSD is used as a swap disk, just like you would do if you want to create a ramdisk, but if you need more capacity, you can store those on a local SSD.

Standard and non-local SSD disks can be sized up to 257 TB for each instance. The performance of these disks scales with each GB of space allocated.

Networking

Robust networking features

- Auto, custom networks
- Inbound/outbound firewall rules
  - IP based
  - Instance/group tags
- Regional HTTPS load balancing
- Network load balancing
  - Does not require pre-warming
- Global and multi-regional subnetworks



As for networking, we have already seen networking features applied to Compute Engine in the previous module's lab. We looked at the different types of networks and created firewall rules using IP addresses and network tags.

You'll also notice that you can do regional HTTPS load balancing and network load balancing. This doesn't require any pre-warming because a load balancer isn't a hardware device that needs to analyze your traffic. A load balancer is essentially a set of traffic engineering rules that are coming into the Google network, and VPC is applying your rules destined to your IP address subnet range. We'll learn more about load balancers in a later course of the Architecting with Google Compute Engine series.

VM access

Linux: SSH

- SSH from the Google Cloud console or Cloud Shell via the Google Cloud SDK
- SSH from computer or third-party client and generate key pair
- Requires firewall rule to allow tcp:22

Windows: RDP

- RDP clients
- Powershell terminal
- Requires setting the Windows password
- Requires firewall rule to allow tcp:3389

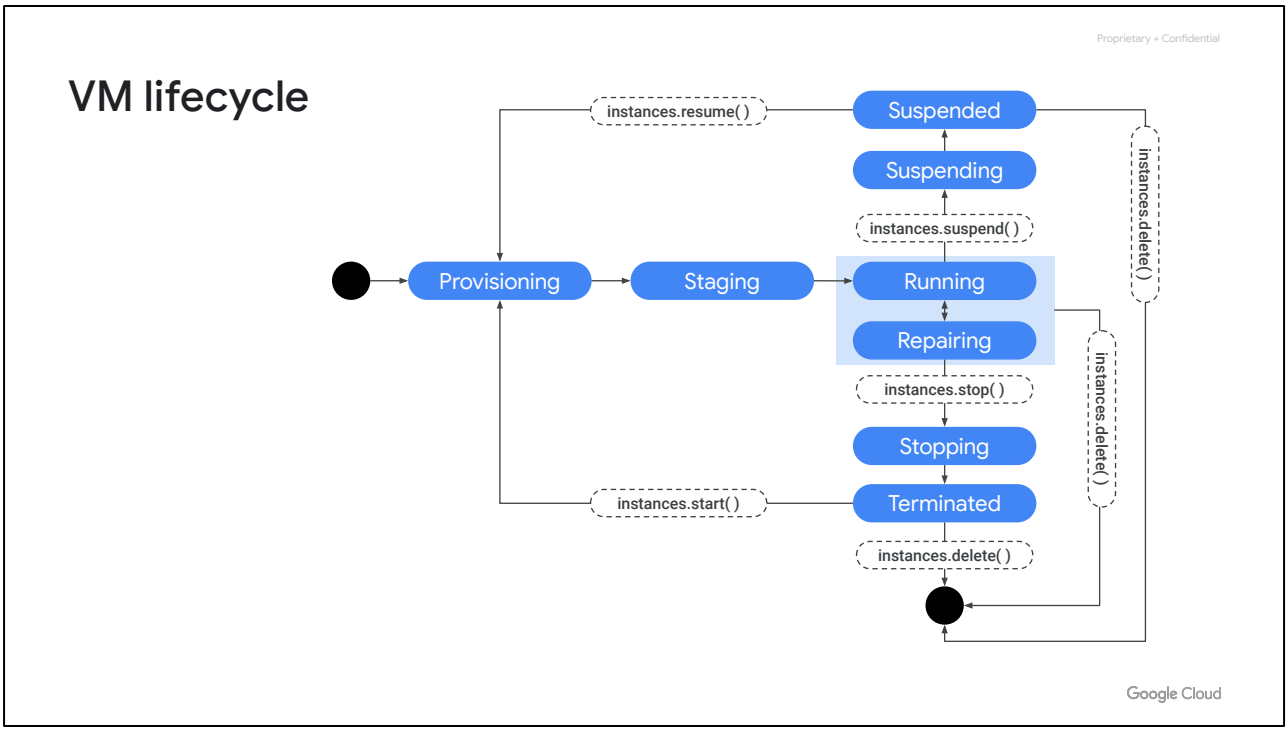
For accessing a VM, the creator of an instance has full root privileges on that instance.

On a Linux instance, the creator has SSH capability and can use the Google Cloud console to grant SSH capability to other users.

On a Windows instance, the creator can use the console to generate a username and password. After that, anyone who knows the username and password can connect to the instance using a Remote Desktop Protocol, or RDP, client.

We listed the required firewall rules for both SSH and RDP here, but you don't need to define these if you are using the default network that we covered in the previous module.

Additional reading:
Choose an access method | Compute Engine Documentation | Google Cloud
Generate credentials for Windows VMs



The lifecycle of a VM is represented by different statuses. We will cover this lifecycle on a high level, but we recommend returning to this diagram as a reference.

When you define all the properties of an instance and click Create, the instance enters the provisioning state. Here the resources such as CPU, memory, and disks are being reserved for the instance, but the instance itself isn't running yet. Next, the instance moves to the staging state where resources have been acquired and the instance is prepared for launch. Specifically, in this state, Compute Engine is adding IP addresses, booting up the system image, and booting up the system.

After the instance starts running, it will go through pre-configured startup scripts and enable SSH or RDP access. Now, you can do several things while your instance is running. For example, you can live migrate your virtual machine to another host in the same zone instead of requiring your instance to be rebooted. This allows Google Cloud to perform maintenance that is integral to keeping the infrastructure protected and reliable, without interrupting any of your VMs. While your instance is running, you can also move your VM to a different zone, take a snapshot of the VM's persistent disk, export the system image, or reconfigure metadata. We will explore some of these tasks in later labs.

Some actions require you to stop your virtual machine; for example, if you want to upgrade your machine by adding more CPU. When the instance enters this state, it will go through pre-configured shutdown scripts and end in the terminated state. From

this state, you can choose to either restart the instance, which would bring it back to its provisioning state, or delete it.

You also have the option to reset a VM, which is similar to pressing the reset button on your computer. This actions wipes the memory contents of the machine and resets the virtual machine to its initial state. The instance remains in the running state through the reset.

The VM may also enter a repairing state. Repairing occurs when the VM encounters an internal error or the underlying machine is unavailable due to maintenance. During this time, the VM is unusable. You are not billed when a VM is in repair. VMs are not covered by the Service level agreement (SLA) while they are in repair. If repair succeeds, the VM returns to one of the above states.

Finally, when you suspend the VM, it enters in the suspending state, before being suspended. You can then resume the VM or delete it.

Changing VM state from running

	Methods	Shutdown Script time	State
`reset`	console, `gcloud`, API, OS	no	remains running
`start`	console, `gcloud`, API	no	terminated➡running
`reboot`	OS: `sudo reboot`	~90 sec	running➡running
`stop`	console, `gcloud`, API	~90 sec	running➡terminated
`shutdown`	OS: `sudo shutdown`	~90 sec	running➡terminated
`delete`	console, `gcloud`, API	~90 sec	running➡N/A
`preemption`	*automatic*	~30 sec	N/A

"ACPI Power Off"

There are different ways you can change a VM state from running. Some methods involve the Google Cloud console and the gcloud command, while others are performed from the OS, such as for reboot and shutdown.

It's important to know that if you are rebooting, stopping, or even deleting an instance, the shutdown process will take about 90 sec. For a preemptible VM, if the instance does not stop after 30 seconds, Compute Engine sends an ACPI G3 Mechanical Off signal to the operating system. Remember that when writing shutdown scripts for preemptible VMs.

Availability policy: Automatic changes

Called "scheduling options" in SDK/API

Automatic restart

- Automatic VM restart due to crash or maintenance event
  - Not preemption or a user-initiated terminate

On host maintenance

- Determines whether host is live-migrated or terminated due to a maintenance event. Live migration is the default.

Live migration

- During maintenance event, VM is migrated to different hardware without interruption.
- Metadata indicates occurrence of live migration.

As I mentioned previously, Compute Engine can live migrate your virtual machine to another host due to a maintenance event to prevent your applications from experiencing disruptions. A VM's availability policy determines how the instance behaves in such an event.

The default maintenance behavior for instances is to live migrate, but you can change the behavior to terminate your instance during maintenance events instead. If your VM is terminated due to a crash or other maintenance event, your instance automatically restarts by default, but this can also be changed.

These availability policies can be configured both during the instance creation and while an instance is running by configuring the Automatic restart and On host maintenance options.

For more information on live migration, refer to the documentation [https://cloud.google.com/compute/docs/instances/live-migration].

Patch management is an essential part of managing an infrastructure

Manage OSes easily through Google Cloud.

- Keep infrastructures up-to-date.
- Reduce the risk of security vulnerabilities.

OS patch management:

- Patch compliance reporting
- Patch deployment

OS updates are a part of managing an infrastructure. Let's see how we can manage the updates to a fleet of Windows VMs.

When you provision a premium image, there is a cost associated with the image. This cost includes both the usage of the OS but also the patch management of the OS. Using Google Cloud, we can easily manage the patching of your OSes.

Managing patches effectively is a great way to keep your infrastructure up-to-date and reduce the risk of security vulnerabilities. But without the right tools, patching can be daunting and labor intensive.

Use OS patch management to apply operating system patches across a set of Compute Engine VM instances. Long-running VMs require periodic system updates to protect against defects and vulnerabilities.

The OS patch management service has two main components:

- Patch compliance reporting, which provides insights on the patch status of your VM instances across Windows and Linux distributions. Along with the insights, you can also view recommendations for your VM instances.
- Patch deployment, which automates the operating system and software patch update process. A patch deployment schedules patch jobs. A patch job runs across VM instances and applies patches.

There are several tasks that can be performed with patch management

Create patch approvals.

Set up flexible scheduling.

Apply advanced patch configuration settings.

Manage these patch jobs or updates from a centralized location.

There are several tasks that can be performed with patch management. You can: …

Create patch approvals. You can select what patches to apply to your system from the full set of updates available for the specific operating system.

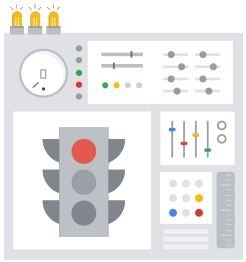
Set up flexible scheduling. You can choose when to run patch updates (one-time and recurring schedules).

Apply advanced patch configuration settings. You can customize your patches by adding configurations such as pre and post patching scripts.

And you can manage these patch jobs or updates from a centralized location.

Charges for stopped (terminated) VMs

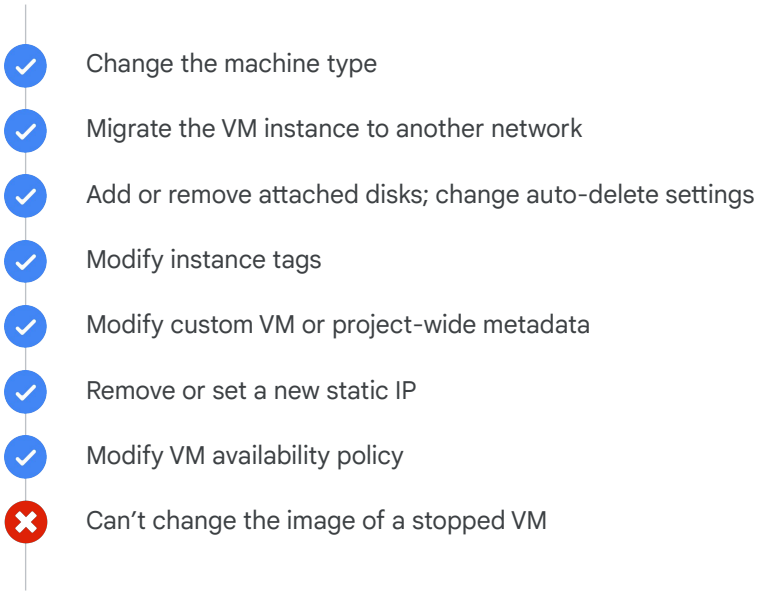
- Memory
- CPU resources



- Attached disks
- Reserved IP addresses

When a VM is terminated, you do not pay for memory and CPU resources. However, you are charged for any attached disks and reserved IP addresses.

Actions supported on a terminated VM

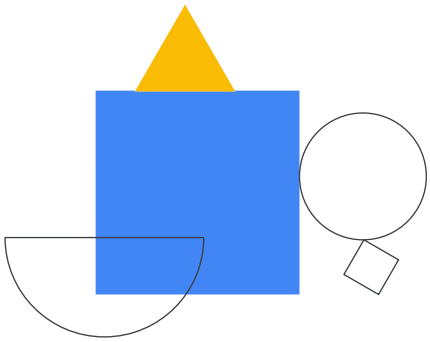


In the terminated state, you can perform any of the actions listed here, such as changing the machine type, but you cannot change the image of a stopped VM.

Also, not all of the actions listed here require you to stop a virtual machine. For example, VM availability policies can be changed while the VM is running, as discussed previously.

Lab Intro

Creating Virtual Machines

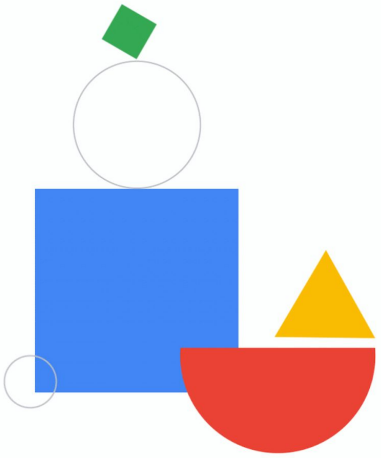


Let’s take some of the Compute Engine concepts that we just discussed and apply them in a lab.

In this lab, you explore virtual machine instance options by creating several standard VMs and a custom VM. You also connect to those VMs using both SSH for Linux machines and RDP for Windows machines.

Lab Review

Creating Virtual Machines



In this lab, you created several virtual machine instances of different types with different characteristics. Specifically, you created a small utility VM for administration purposes, a Windows VM, and a custom Linux VM. You also accessed both the Windows and Linux VMs and deleted all your created VMs.

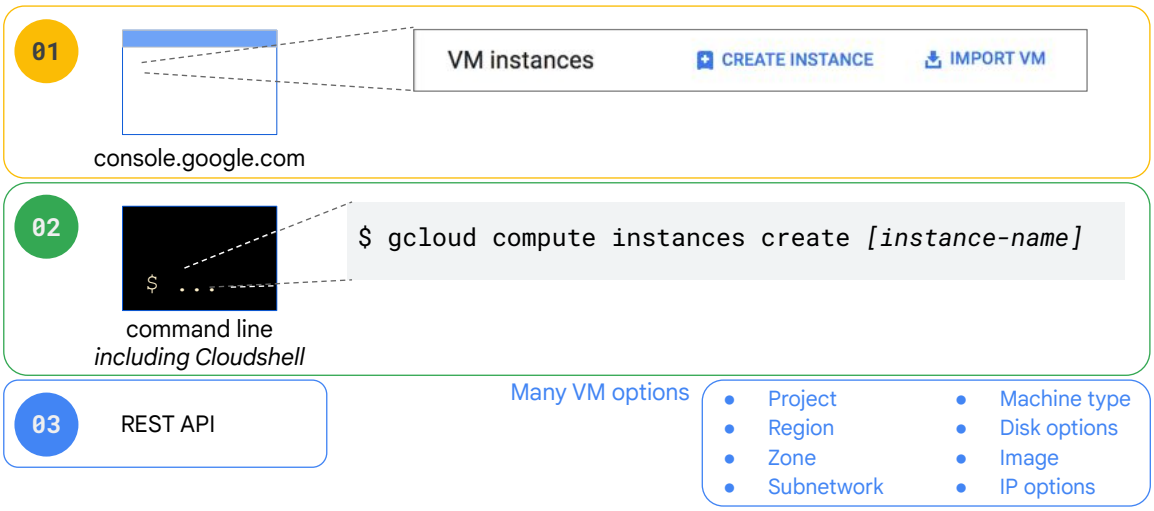
In general, start with smaller VMs when you're prototyping solutions to keep the costs down. When you are ready for production, trade up to larger VMs based on capacity. If you're building in redundancy for availability, remember to allocate excess capacity to meet performance requirements. Finally, consider using custom VMs when your application's requirements fit between the features of the standard types.

You can stay for a lab walkthrough, but remember that Google Cloud's user interface can change, so your environment might look slightly different.

Compute Options (vCPU and Memory)

Now that you have completed the lab, let's dive deeper into the compute options that are available to you in Google Cloud, by focusing on CPU and memory.

Creating a VM

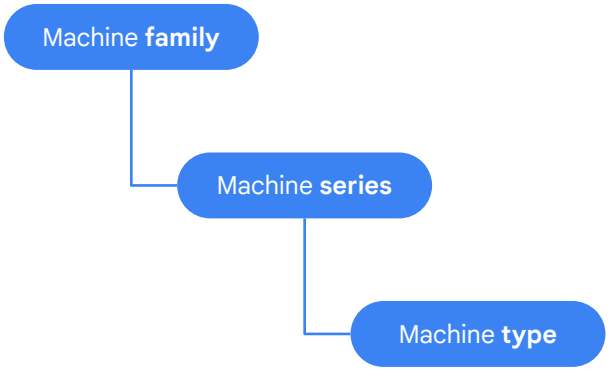


You have three options for creating and configuring a VM. You can use the Cloud Console as you did in the previous lab, the Cloud Shell command line, or the RESTful API. If you'd like to automate and process very complex configurations, you might want to programmatically configure these through the RESTful API by defining all the different options for your environment.

If you plan on using the command line or RESTful API, I recommend that you first configure the instance through the Cloud Console and then ask Compute Engine for the equivalent REST request or command line, as shown in the demo earlier. This way you avoid any typos and get dropdown lists of all the available CPU and memory options.

Speaking of CPU and memory options, let's look at the different machine types that are currently available.

Machine type structure



When you create a VM, you select a machine type from a machine family that determines the resources available to that VM. There are several machine families you can choose from and each machine family is further organized into machine series and predefined machine types within each series.

A machine family is a curated set of processor and hardware configurations optimized for specific workloads. When you create a VM instance, you choose a predefined or custom machine type from your preferred machine family.

Alternatively, you can create custom machine types. These let you specify the number of vCPUs and the amount of memory for your instance.

Compute Engine machine families

- General-purpose
- Compute-optimized
- Memory-optimized
- Accelerator-optimized

There are four Compute Engine machine families.

- General-purpose
- Compute-optimized
- Memory-optimized, and
- Accelerator-optimized

Let’s look at each in more detail.

General-purpose machine family

Series	Workload	Applications	
E2	(Cost-optimized) Day-to-day computing at a lower cost	• Web serving • App serving • Back office apps • Small-medium databases	• Microservices • Virtual desktops • Development environments
N2, N2D, N1	(Balanced) Balanced price/performance across a wide range of VM shapes	• Web serving • App serving • Back office apps	• Medium-large databases • Cache • Media/streaming
Tau T2D, Tau T2A	(Scale-out optimized) Best performance/ cost for scale-out workloads	• Scale-out workloads • Web serving • Containerized microservices	• Media transcoding • Large-scale Java applications

The general-purpose machine family has the best price-performance with the most flexible vCPU to memory ratios, and provides features that target most standard and cloud-native workloads.

The E2 machine series is suited for day-to-day computing at a lower cost, especially where there are no application dependencies on a specific CPU architecture.

E2 VMs provide a variety of compute resources for the lowest price on Compute Engine, especially when paired with committed-use discounts. You simply pick the amount of vCPU and memory you want, and Google provisions it for you. The Standard E2 VMs have between 2 to 32 vCPUs with a ratio of 0.5 GB to 8 GB of memory per vCPU.

They are a great choice for web servers, small to medium databases, development and test environments, and many applications that don't have strict performance requirements. They offer a compatible performance baseline with the current N1 VMs for those of you who have been using them.

The E2 machine series also contains shared-core machine types that use context-switching to share a physical core between vCPUs for multitasking. Different shared-core machine types sustain different amounts of time on a physical core. In general, shared-core machine types can be more cost-effective for running small, non-resource intensive applications than standard, high-memory, or high-CPU machine types. Shared-core E2 machine types have 0.25 to 1 vCPUs with 0.5 GB to

8 GB of memory.

The N2 and N2D are the next generation following the N1 VMs, offering a significant performance jump.

N2 and N2D are the most flexible VM types and provide a balance between price and performance across a wide range of VM shapes, including enterprise applications, medium-to-large databases, and many web and app-serving workloads. Committed use and sustained use discounts are supported.

N2 VMs support the latest second generation scalable processor from Intel with up to 128 vCPUs and 0.5 to 8 GB of memory per vCPU. Cascade Lake is the default processor for machine types up to 80 vCPUs. For larger machine types Ice Lake is the default processor for specific regions and zones.

N2D are AMD-based general purpose VMs. They leverage the latest EPYC Milan and EPYC Rome processors, and provide up to 224 vCPUs per node.

Tau T2D and Tau T2A VMs are optimized for cost-effective performance of demanding scale-out workloads.

T2D VMs are built on the latest 3rd Gen AMD EPYCTM processors and offer full x86 compatibility. They are suited to scale-out workloads including web servers, containerized microservices, media transcoding, and large-scale Java applications. T2D VMs come in predefined VM shapes, with up to 60 vCPUs per VM and 4 GB of memory per vCPU.

Tau T2A machine series is the first machine series in Google Cloud to run on Arm processors. The Tau T2A machine series runs on a 64 core Ampere Altra processor with an Arm instruction set and an all-core frequency of 3 GHz.

If you have containerized workloads, Tau VMs are supported by Google Kubernetes Engine to help optimize price-performance. You can add T2D nodes to your GKE clusters by specifying the T2D machine type in your GKE node pools.

General purpose machines:
https://cloud.google.com/compute/docs/general-purpose-machines

Compute-optimized machine family

Series	Workload	Applications	
C2	Ultra high performance for compute-intensive workloads	• Compute-bound workloads • High-performance web serving • Gaming (AAA game servers) • Ad serving	• High-performance computing (HPC) • Media transcoding • AI/ML
C2D	Ultra high performance for compute-intensive workloads	• Memory-bound workloads • Gaming (AAA game servers) • High-performance computing (HPC)	• High-performance databases • Electronic Design Automation (EDA) • Media transcoding
H3	Ultra high performance for compute-intensive workloads	• High-performance computing (HPC)	• Electronic Design Automation (EDA)

The compute-optimized machine family has the highest performance per core on Compute Engine and is optimized for compute-intensive workloads.

C2 VMs are the best fit VM type for compute-intensive workloads, including AAA gaming, electronic design automation, and high-performance computing across simulations, genomic analysis, or media transcoding. They might also be applications that have very expensive per core licensing and thus would benefit from higher per core performance.

Powered by high-frequency Intel-scalable processors, Cascade Lake, C2 machine types offer up to 3.8 Ghz sustained all-core turbo and provide full transparency into the architecture of the underlying server platforms, enabling advanced performance tuning.

The C2 series comes in different machine types ranging from 4 to 60 vCPUs, and offers up to 240 GB of memory. You can also attach up to 3 TB of local storage to these VMs for applications that require higher storage performance.

The C2D machine series provides the largest VM sizes and are best-suited for high-performance computing (HPC). The C2D series also has the largest available last-level cache (LLC) cache per core.

The C2D machine series come in different machine types ranging from 2 to 112 vCPUs, and offer 4 GB of memory per vCPU. You can also attach up to 3TB of local

storage to these machine types for applications that require higher storage performance. C2D VMs are available on the third generation AMD EPYC Milan platform.

The H3 series offer 88 cores and 352 GB of DDR5 memory and are available on the Intel Sapphire Rapids CPU platform and Google's custom Intel Infrastructure Processing Unit (IPU).

[Compute-optimized machines: https://cloud.google.com/compute/docs/compute-optimized-machines]

Memory-optimized machine family

Series	Workload	Applications	
M1	Ultra high-memory workloads	• Medium in-memory databases such as SAP HANA • Tasks that require intensive use of memory with higher memory-to-vCPU ratios than the general-purpose high-memory machine types.	• In-memory databases and in-memory analytics, business warehousing (BW) workloads, genomics analysis, SQL analysis services. • Microsoft SQL Server and similar databases.
M2	Ultra high-memory workloads	• Large in-memory databases such as SAP HANA	• In-memory databases and in-memory analytics, business warehousing (BW) workloads, genomics analysis, SQL analysis services, etc.
M3	Ultra high-memory workloads	• OLAP and OLTP SAP workloads	• Memory Intense Electronic Design Automation

The memory-optimized machine family provides the most compute and memory resources of any Compute Engine machine family offering. They are ideal for workloads that require higher memory-to-vCPU ratios than the high-memory machine types in the general-purpose machine family.

The M1 machine series has up to 4 TB of memory, while the M2 machine series has up to 12 TB of memory. These machine series are well-suited for large in-memory databases such as SAP HANA, as well as in-memory data analytics workloads.

Both the M1 and M2 machine series offer the lowest cost per GB of memory on Compute Engine, making them a great choice for workloads that utilize higher memory configurations with low compute resources requirements. Additionally, they offer up to 30% sustained use discounts and are also eligible for committed use discounts, bringing additional savings of greater than 60% for three-year commitments.

M3 VMs offer up to 128 vCPUs, with up to 30.5 GB of memory per vCPU, and are available on the Intel Ice Lake CPU platform. These machines are well-suited for memory intensive applications, such as genomic modeling and electronic design automation and high performance computing.

[Memory-optimized machines: https://cloud.google.com/compute/docs/memory-optimized-machines]

Accelerator-optimized machine family

Series	Workload	Applications
A2	Optimized for high-performance computing workloads	• CUDA-enabled ML training and inference • HPC • Massive parallelized computation
G2	Optimized for high-performance computing workloads	• Video transcoding • Remote visualization workstation

The accelerator-optimized machine family is ideal for massively parallelized Compute Unified Device Architecture (CUDA) compute workloads, such as machine learning (ML) and high-performance computing (HPC). This family is the optimal choice for workloads that require GPUs.

The A2 series has 12 to 96 vCPUs, and up to 1360 GB of memory. Each A2 machine type has a fixed number (up to 16) of NVIDIA's Ampere A100 GPUs attached. An A100 GPU provides 40 GB of GPU memory—ideal for large language models, databases, and HPC.

G2 VMs offer 4 to 96 vCPUs, up to 432 GB of memory, and are available on the Intel Cascade Lake CPU platform. These machines are well-suited for CUDA-enabled ML training and inference, video transcoding, remote visualization workstation.

[Accelerator-optimized machine family: https://cloud.google.com/compute/docs/accelerator-optimized-machines]

Specs for currently available VM machine types:

cloud.google.com/compute/docs/machine-types

Additional information, including the latest specs for currently available VM machine types, can be found in the machine types documentation.

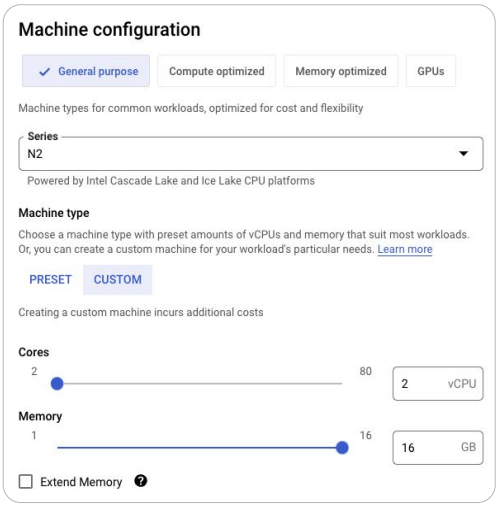
Creating custom machine types

When to select custom:

- Requirements fit between the predefined types
- Need more memory or more CPU

Customize the amount of memory and vCPU for your machine:

- Either 1 vCPU or even number of vCPU
- Up to 8 GB per vCPU
- Total memory must be multiple of 256 MB



Google Cloud

If none of the predefined machine types match your needs, you can independently specify the number of vCPUs and the amount of memory for your instance. Custom machine types are ideal for the following scenarios:

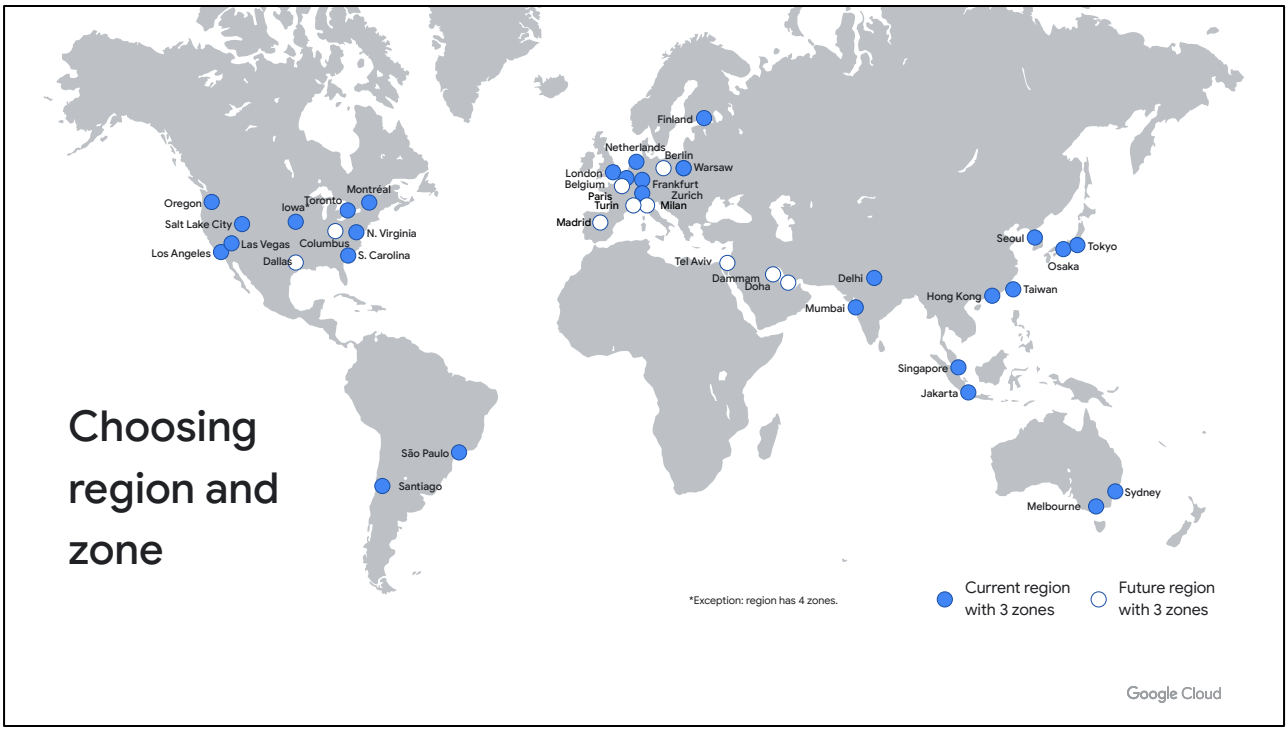
- When you have workloads that are not a good fit for the predefined machine types that are available to you.
- Or when you have workloads that require more processing power or more memory, but don't need all of the upgrades that are provided by the next larger predefined machine type.

It costs slightly more to use a custom machine type than an equivalent predefined machine type, and there are still some limitations in the amount of memory and vCPUs you can select:

- Only machine types with 1 vCPU or an even number of vCPUs can be created.
- Memory must be between 1 GB and 8 GB per vCPU.
- The total memory of the instance must be a multiple of 256 MB.

Selected custom machine types can allow up to 8 GB of memory per vCPU. However, this might not be enough memory for your workload. At an additional cost, you can get more memory per vCPU beyond the 8 GB limit. This is referred to as *extended memory,* and you can learn more about this in the link provided in the module PDF located in Course Resources.

[https://cloud.google.com/compute/docs/instances/creating-instance-with-custom-machine-type#extendedmemory]



The first thing you want to consider when choosing a region and zone is the geographical location in which you want to run your resources. This map shows the current and planned Google Cloud regions and number of zones. For up-to-date information on the available regions and zones, see the documentation linked in the module PDF located in Course Resources.

Each zone supports a combination of Ivy Bridge, Sandy Bridge, Haswell, Broadwell, and Skylake platforms. When you create an instance in the zone, your instance will use the default processor supported in that zone. For example, if you create an instance in the us-central1-a zone, your instance will use a Sandy Bridge processor.

[Regions and zones:
https://cloud.google.com/compute/docs/regions-zones/#available]

Pricing

- Per-second billing, with minimum of 1 minute
  - vCPUs, GPUs, and GB of memory
- Resource-based pricing:
  - Each vCPU and each GB of memory is billed separately
- Discounts:
  - Sustained use
  - Committed use
  - Preemptible and Spot VM instances
- Recommendation Engine
  - Notifies you of underutilized instances
- Free usage limits

Google Cloud offers a variety of different options to keep the prices low for Compute Engine resources:

- All vCPUs, GPUs, and GB of memory are charged a minimum of 1 minute. For example, if you run your virtual machine for 30 seconds, you will be billed for 1 minute of usage. After 1 minute, instances are charged in 1-second increments.
- Compute Engine uses a resource-based pricing model, where each vCPU and each GB of memory on Compute Engine is billed separately rather than as part of a single machine type. You still create instances using predefined machine types, but your bill reports them as individual vCPUs and memory used.
- There are several discounts available but the discount types cannot be combined:
  - Resource-based pricing allows Compute Engine to apply sustained use discounts to all of your predefined machine type usage in a region collectively rather than to individual machine types.
  - If your workload is stable and predictable, you can purchase a specific amount of vCPUs and memory for a discount off of normal prices in return for committing to a usage term of 1 year or 3 years. The discount is up to 57% for most machine types or custom machine types. The discount is up to 70% for memory-optimized machine types.
  - Preemptible and Spot VMs are instances that you can create and run at a much lower price than normal instances. For both types of VM,

    - Compute Engine might terminate (or preempt) these instances if it requires access to those resources for other tasks. Both preemptible VMs and Spot VMs are excess Compute Engine capacity so their availability varies with usage. Importantly, preemptible VMs can only run for up to 24 hours at a time, but Spot VMs do not have a maximum runtime.
- The ability to customize the amount of memory and CPU through custom machine types allows for further pricing customization. Speaking of sizing your machine, Compute Engine provides VM sizing recommendations to help you optimize the resource used of your virtual machine instances. When you create a new instance, recommendations for the new instance will appear 24 hours after the instance has been created.
- Compute Engine also has Free Usage Limits. [https://cloud.google.com/free/docs/gcp-free-tier#always-free-usage-limits]

Sustained use discounts

Sustained use discounts for up to 30%

General-purpose N1 predefined and custom machine types, memory-optimized machine types, shared-core machine types, and sole-tenant nodes

Usage level (% of month)	% at which incremental is charged
0% - 25%	100% of base rate
25% - 50%	80% of base rate
50% - 75%	60% of base rate
75% - 100%	40% of base rate

Sustained use discounts for up to 20%

General-purpose N2 and N2D predefined and custom machine types, and compute-optimized machine types

Usage level (% of month)	% at which incremental is charged
0% - 25%	100% of base rate
25% - 50%	86.78% of base rate
50% - 75%	73.3% of base rate
75% - 100%	60% of base rate

Up to 30% net discount for instances that run the entire month

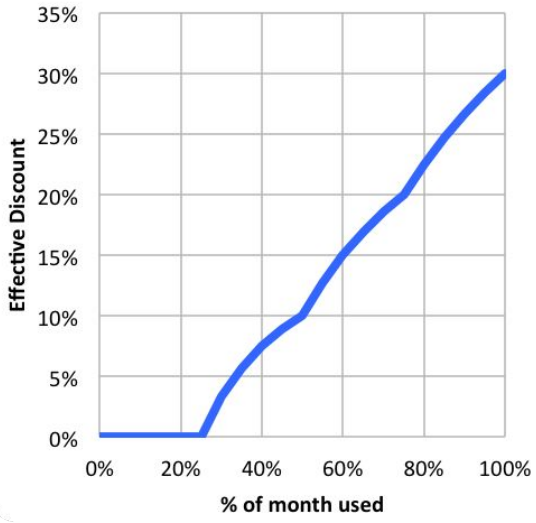
Sustained use discounts are automatic discounts that you get for running specific Compute Engine resources (vCPUs, memory, GPU devices) for a significant portion of the billing month. For example, when you run one of these resources for more than 25% of a month, Compute Engine automatically gives you a discount for every incremental minute you use for that instance. The discount increases with usage, and you can get up to a 30% net discount for instances that run the entire month.

The tables shown on this slide describes the discount you get at each usage level of a VM instance. To take advantage of the full 30% discount, create your VM instances on the first day of the month, because discounts reset at the beginning of each month.

[Sustained use discounts: https://cloud.google.com/compute/docs/sustained-use-discounts]

Sustained use discounts increase with use

cloud.google.com/products/calculator

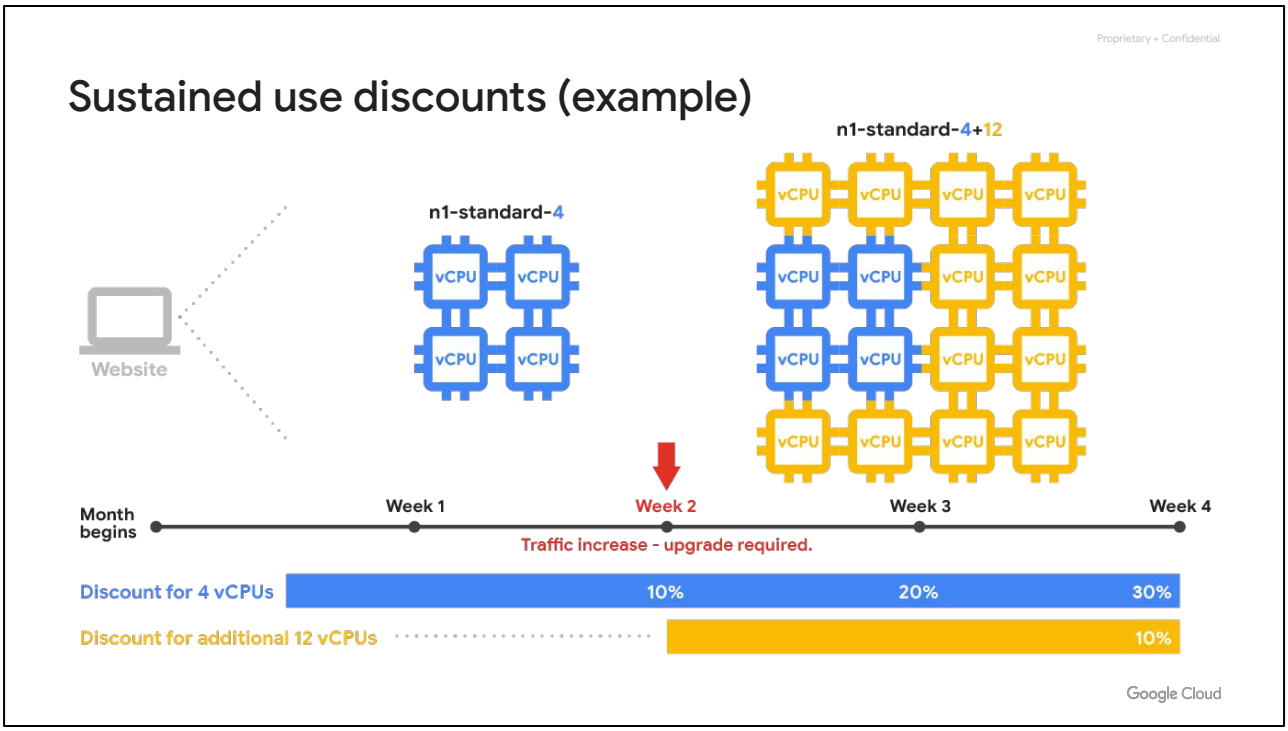


The graph on this slide demonstrates how your effective discount increases with use. For example, if you use a virtual machine for 50% of the month, you get an effective discount of 10%. If you use it for 75% of the month, you get an effective discount of 20%. If you use it for 100% of the month, you get an effective discount of 30%. You can also use the Google Cloud Pricing Calculator to estimate your sustained use discount for any arbitrary workload.

Compute Engine calculates sustained use discounts based on vCPU and memory usage across each region and separately for each of the following categories:

- Predefined machine types
- Custom machine type

Google Cloud Pricing Calculator: [https://cloud.google.com/products/calculator/]

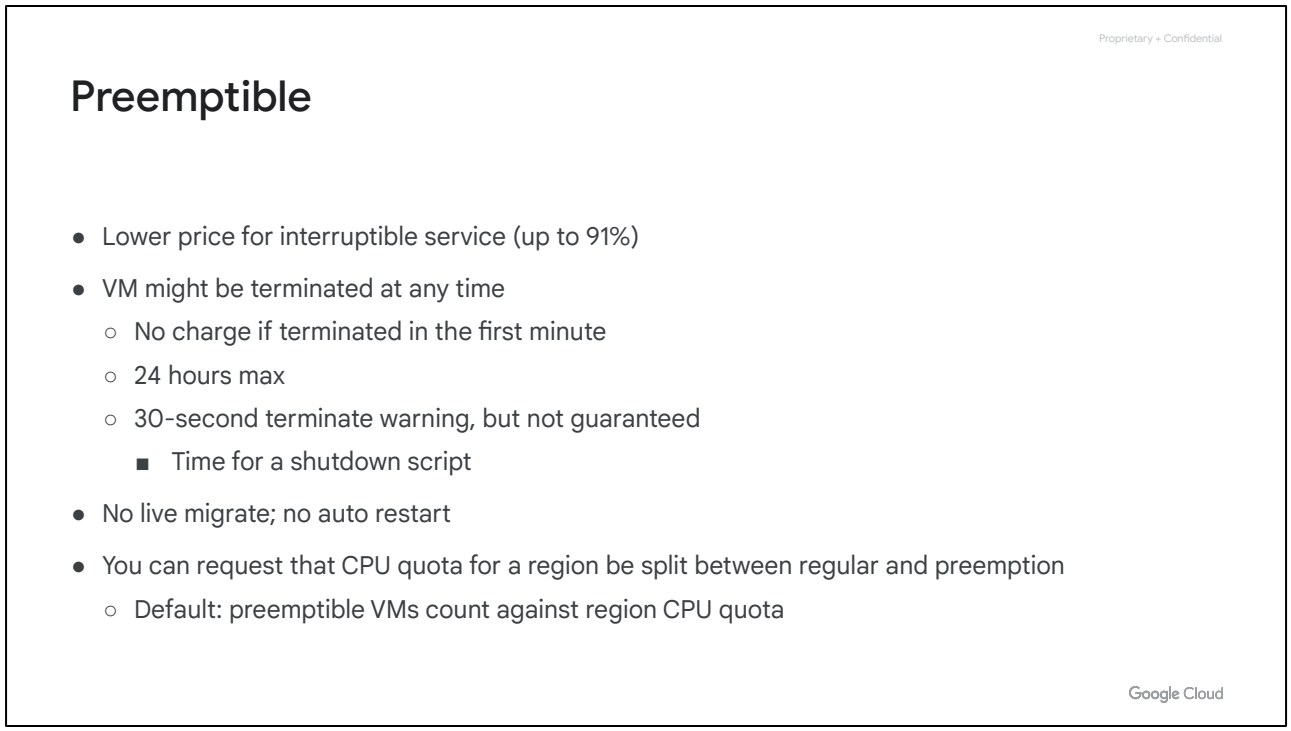


Let's go through an example where you have two instances that are in the same region but have different machine types and run at different times of the month. Compute Engine breaks down the number of vCPUs and amount of memory used across all instances that use predefined machine types and combines the resources to qualify for the largest sustained usage discounts possible. As shown on this slide, you run the following two instances in the us-central1 region during a month:

- For the first half of the month, you run an n1-standard-4 instance with 4 vCPUs and 15 GB of memory
- For the second half of the month, you run a larger n1-standard-16 instance with 16 vCPUs and 60 GB of memory

In this scenario, Compute Engine reorganizes these machine types into individual vCPUs and memory resources and combines their usage to create the following resources, as shown on the bottom:

- 4 vCPUs and 15 GB of memory for a full month
- 12 vCPUs and 45 GB of memory for half of the month



As I mentioned earlier, a preemptible VM is an instance that you can create and run at a much lower cost than normal instances.

See whether you can make your application function completely on preemptible VMs, because an 60-91% discount is a significant investment in your application.

Now, just to reiterate, these VMs might be preempted at any time, and there is no charge if that happens within the first minute. Also, preemptible VMs are only going to live for up to 24 hours, and you only get a 30-second notification before the machine is preempted.

It's also worth noting that there are no live migrations nor automatic restarts in preemptible VMs, but something that we will highlight is that you can actually create monitoring and load balancers that can start up new preemptible VMs in case of a failure. In other words, there are external ways to keep restarting preemptible VMs if you need to.

One major use case for preemptible VMs is running a batch processing job. If some of those instances terminate during processing, the job slows but does not completely stop. Therefore, preemptible instances complete your batch processing tasks without placing additional workload on your existing instances, and without requiring you to pay full price for additional normal instances.

[https://cloud.google.com/compute/docs/instances/preemptible#what_is_a_preemptibl

e_instance]

Spot VMs

- Spot VMs are the latest version of preemptible VMs
- Spot VMs and preemptible VMs share the same pricing model
- No minimum or maximum runtime
- Spot VMs are finite Compute Engine resources, so they might not always be available
- No live migrate; no auto restart
- Best practice use cases help you get the most of using Spot VMs

Spot VMs are the latest version of preemptible VMs. Spot VMs are virtual machine (VM) instances with the spot provisioning model. New and existing preemptible VMs continue to be supported, and preemptible VMs use the same pricing model as Spot VMs. However, Spot VMs provide new features that preemptible VMs do not support. For example, preemptible VMs can only run for up to 24 hours at a time, but Spot VMs do not have a maximum runtime.

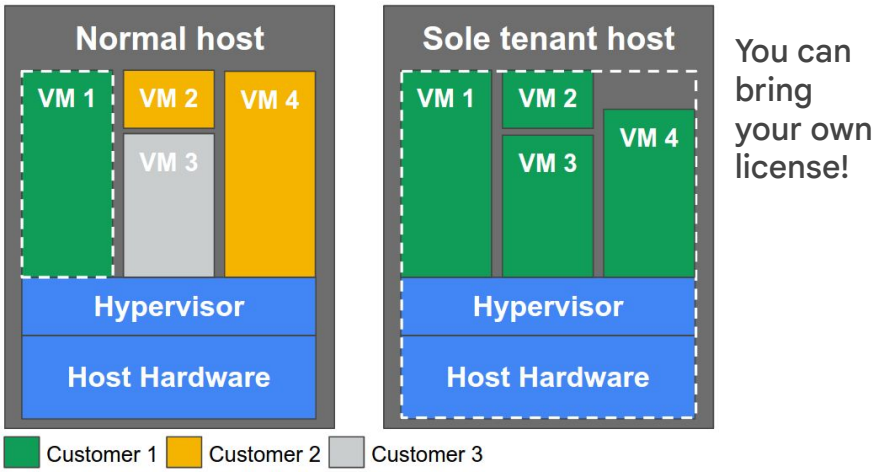
Like preemptible VMs, Compute Engine might preempt Spot VMs if it needs to reclaim those resources for other tasks. The probability that Compute Engine stops Spot VMs for a system event is generally low, but might vary from day to day and from zone to zone depending on current conditions. Spot VMs are finite Compute Engine resources, so they might not always be available. Like preemptible VMs, it’s worth noting that Spot VMs can't live-migrate to become standard VMs while they are running or be set to automatically restart when there is a maintenance event.

There are many best practices which can help you get the most of using Spot VMs. For example, resources for Spot VMs come out of excess and backup Google Cloud capacity. Capacity for Spot VMs is often easier to get for smaller machine types, meaning machine types with less resources like vCPUs and memory.

[For more information on best practices, see

https://cloud.google.com/compute/docs/instances/create-use-spot#best-practices]
[For more information on Spot VMs, see
https://cloud.google.com/compute/docs/instances/spot]

Sole-tenant nodes physically isolate workloads



If you have workloads that require physical isolation from other workloads or virtual machines in order to meet compliance requirements, you want to consider sole-tenant nodes.

A sole-tenant node is a physical Compute Engine server that is dedicated to hosting VM instances only for your specific project. Use sole-tenant nodes to keep your instances physically separated from instances in other projects, or to group your instances together on the same host hardware, for example if you have a payment processing workload that needs to be isolated to meet compliance requirements.

The diagram on the left shows a normal host with multiple VM instances from multiple customers. A sole tenant node as shown on the right also has multiple VM instances, but they all belong to the same project. You can also fill the node with multiple smaller VM instances of various sizes, including custom machine types and instances with extended memory. Also, if you have existing operating system licenses, you can bring them to Compute Engine using sole-tenant nodes while minimizing physical core usage with the in-place restart feature.

[https://cloud.google.com/compute/docs/nodes/create-nodes]

[For more information on sole tenancy and allowed node types, please refer to the sole-tenancy overview in the documentation.]

Shielded VMs offer verifiable integrity

- Secure Boot
- Virtual trusted platform module (vTPM)
- Integrity monitoring

Requires shielded image!

Another compute option is to create shielded VMs. Shielded VMs offer verifiable integrity of your VM instances, so you can be confident that your instances haven't been compromised by boot- or kernel-level malware or rootkits.

Shielded VM's verifiable integrity is achieved through the use of Secure Boot, virtual trusted platform module or vTPM-enabled Measured Boot, and integrity monitoring.

Shielded VMs is the first offering in the Shielded Cloud initiative. The Shielded Cloud initiative is meant to provide an even more secure foundation for all of Google Cloud by providing verifiable integrity and offering features, like vTPM shielding or sealing, that help prevent data exfiltration.

To use these shielded VM features, you need to select a shielded image. We'll learn about images in the next section.

Confidential VMs allow you to encrypt data in use

- Encrypts data while it’s being processed.
- Easy to use with no changes to code or performance compromise.
- N2D Compute Engine VM running on second generation AMD Epyc processors.
- Provides high memory capacity, high throughput, and supports parallel and compute heavy workloads.
- You can select Confidential VM service when creating a new VM.

Confidential VMs are a breakthrough technology that allows you to encrypt data in use—while it’s being processed.

Google Cloud’s approach to encrypt data in use is simple, easy-to-use deployment without making any code changes to their applications or having to compromise on performance. You can collaborate with anyone, all while preserving the confidentiality of your data.

A Confidential Virtual Machine (Confidential VM) is a type of N2D Compute Engine VM running on hosts based on the second generation of AMD Epyc processors, code-named "Rome."

Using AMD Secure Encrypted Virtualization (SEV), Confidential VM features built-in optimization of both performance and security for enterprise-class high memory workloads, as well as inline memory encryption that doesn't introduce significant performance penalty to those workloads.

The AMD Rome processor family is specifically optimized for compute-heavy workloads, with high memory capacity, high throughput, and support for parallel workloads. In addition, AMD SEV provides for Confidential Computing support. With the confidential execution environments provided by Confidential VM and AMD SEV, Google Cloud keeps customers' sensitive code and other data encrypted in memory during processing. Google does not have access to the encryption keys.

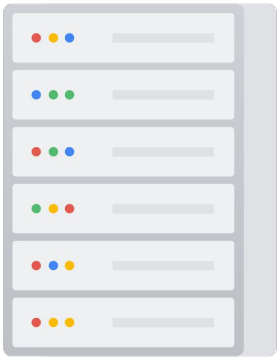
You can select the Confidential VM service when creating a new VM using the Google Cloud Console, the Compute Engine API, or the gcloud command-line tool.

Images

Next, let's focus on images.

What's in an image?

- Boot loader
- Operating system
- File system structure
- Software
- Customizations



When creating a virtual machine, you can choose the boot disk image. This image includes the boot loader, the operating system, the file system structure, any pre-configured software, and any other customizations.

Images

Public base images

- Google, third-party vendors, and community; Premium images (p)
- Linux
  - CentOS, CoreOS, Debian, RHEL(p), SUSE(p), Ubuntu, openSUSE, and FreeBSD
- Windows
  - Windows Server 2019(p), 2016(p), 2012-r2(p)
  - SQL Server pre-installed on Windows(p)

Custom images

- Create new image from VM: pre-configured and installed SW
- Import from on-prem, workstation, or another cloud
- Management features: image sharing, image family, deprecation

You can select either a public or custom image.

As you saw in the previous lab, you can choose from both Linux and Windows images. Some of these images are premium images, as indicated in parentheses with a p. These images will have per-second charges after a 1-minute minimum, with the exception of SQL Server images, which are charged per minute after a 10-minute minimum. Premium image prices vary with the machine type. However, these prices are global and do not vary by region or zone.

You can also use custom images. For example, you can create and use a custom image by pre-installing software that's been authorized for your particular organization.

You also have the option of importing images from your own premises or workstation, or from another cloud provider. This is a no-cost service that is as simple as installing an agent, and I highly recommend that you look at it. You can also share custom images with anybody in your project or among other projects, too.

Machine images

Scenarios	Machine image	Persistent disk snapshot	Custom image	Instance template
Single disk backup	Yes	Yes	Yes	No
Multiple disk backup	Yes	No	No	No
Differential backup	Yes	Yes	No	No
Instance cloning	Yes	No	Yes	Yes
Base image replication	No	No	Yes	No

A machine image is a Compute Engine resource that stores all the configuration, metadata, permissions, and data from one or more disks required to create a virtual machine (VM) instance. You can use a machine image in many system maintenance scenarios, such as creation, backup and recovery, and instance cloning.

Machine images are the most ideal resources for disk backups as well as instance cloning and replication.

Disk Options

At this point you've chosen an operating system, but that operating system is going to be included as part of some kind of disk. So let’s look at the disk options.

Boot disk

- VM comes with a single root persistent disk.
- Image is loaded onto root disk during first boot:
  - Bootable: you can attach to a VM and boot from it.
  - Durable: can survive VM terminate.
- Some OS images are customized for Compute Engine.
- Can survive VM deletion if “Delete boot disk when instance is deleted” is disabled.

Every single VM comes with a single root persistent disk, because you're choosing a base image to have that loaded on.

This image is bootable in that you can attach it to a VM and boot from it, and it is durable in that it can survive if the VM terminates. To have a boot disk survive a VM deletion, you need to disable the “Delete boot disk when instance is deleted” option in the instance’s properties.

As I discussed earlier, there are different types of disks. Let’s explore these in more detail.

Persistent disks

Network storage appearing as a block device

- Attached to a VM through the network interface
- Durable storage: can survive VM terminate
- Bootable: you can attach to a VM and boot from it
- Snapshots: incremental backups
- Performance: Scales with size
- HDD (magnetic) or SSD (faster, solid-state) options
- Disk resizing: even running and attached!
- Can be attached in read-only mode to multiple VMs
- Zonal or Regional
  - `pd-standard`
  - `pd-ssd`
  - `pd-balanced`
  - `pd-extreme` (zonal only)
- Encryption keys:
  - Google-managed
  - Customer-managed
  - Customer-supplied

The first disk that we create is what we call a persistent disk. That means it's going to be attached to the VM through the network interface. Even though it's persistent, it's not physically attached to the machine. This separation of disk and compute allows the disk to survive if the VM terminates. You can also perform snapshots of these disks, which are incremental backups that we'll discuss later.

The choice between HDD and SSD disks comes down to cost and performance. To learn more about disk performance and how it scales with disk size, refer to the documentation https://cloud.google.com/compute/docs/disks/performance.

Another cool feature of persistent disks is that you can dynamically resize them, even while they are running and attached to a VM.

You can also attach a disk in read-only mode to multiple VMs. This allows you to share static data between multiple instances, which is cheaper than replicating your data to unique disks for individual instances.

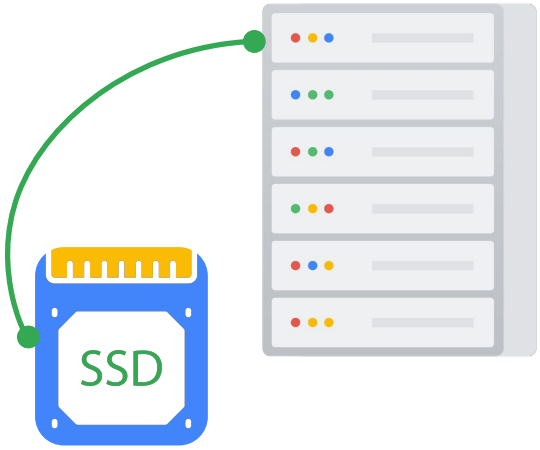
Zonal persistent disks offer efficient, reliable block storage. Regional persistent disks provide active-active disk replication across two zones in the same region. Regional persistent disks deliver durable storage that is synchronously replicated across zones and are a great option for high-performance databases and enterprise applications that also require high availability. When you configure a zonal or regional persistent disk, you can select one of the following disk types.

- Standard persistent disks (`pd-standard`). These types of disks are backed by standard hard disk drives (HDD) and are suitable for large data processing workloads that primarily use sequential I/Os.
- Performance SSD persistent disks (`pd-ssd`). These types of disks are backed by solid-state drives (SSD) and are suitable for enterprise applications and high-performance databases that require lower latency and more IOPS than standard persistent disks provide.
- Balanced persistent disks (`pd-balanced`). These types of disks are also backed by solid-state drives (SSD). They are an alternative to SSD persistent disks that balance performance and cost. These disks have the same maximum IOPS as SSD persistent disks and lower IOPS per GB. For most VM shapes, except very large ones, this disk type offers performance levels suitable for most general-purpose applications at a price point between that of standard and performance (pd-ssd) persistent disks.
- Extreme persistent disks (`pd-extreme`) are zonal persistent disks and are also backed by solid-state drives (SSD). Extreme persistent disks are designed for high-end database workloads, providing consistently high performance for both random access workloads and bulk throughput. Unlike other disk types, you can provision your desired IOPS.

By default, Compute Engine encrypts all data at rest. Google Cloud handles and manages this encryption for you without any additional actions on your part. However, if you wanted to control and manage this encryption yourself, you can either use Cloud Key Management Service to create and manage key encryption keys (which is known as customer-managed encryption keys) or create and manage your own key encryption keys (known as customer-supplied encryption keys).

Local SSD disks are physically attached to a VM

- More IOPS, lower latency, and higher throughput than persistent disk
- 375-GB disk up to 24, total of 9 TB
- Data survives a reset, but not a VM stop or terminate
- VM-specific: cannot be reattached to a different VM



Now, local SSDs are different from persistent disks in that they are physically attached to the virtual machine. Therefore, these disks are ephemeral but provide very high IOPS. For up-to-date numbers I recommend referring to the documentation [https://cloud.google.com/compute/docs/disks/local-ssd#performance].

Currently, you can attach up to 24 local SSD disks with 375 GB each, resulting in a total of 9 TB.

Data on these disks will survive a reset but not a VM stop or terminate, because these disks can't be reattached to a different VM.

RAM disk

- `tmpfs`
- Faster than local disk, slower than memory
  - Use when your application expects a file system structure and cannot directly store its data in memory
  - Fast scratch disk, or fast cache
- Very volatile; erase on stop or restart
- May need a larger machine type if RAM was sized for the application
- Consider using a persistent disk to back up RAM disk data

You also have the option of using a RAM disk.

You can simply use tmpfs if you want to store data in memory. This will be the fastest type of performance available if you need small data structures. I recommend a high-memory virtual machine if you need to take advantage of such features, along with a persistent disk to back up the RAM disk data.

Summary of disk options

	Persistent disk HDD	Persistent disk SSD	Local SSD disk	RAM disk
Data redundancy	Yes	Yes	No	No
Encryption at rest	Yes	Yes	Yes	N/A
Snapshotting	Yes	Yes	No	No
Bootable	Yes	Yes	No	Not
Use case	General, bulk file storage	Very random IOPS	High IOPS and low latency	low latency and risk of data loss

In summary, you have several different disk options. Persistent disks can be rebooted and snapshotted, but local SSDs and RAM disks are ephemeral.

I recommend choosing a persistent HDD disk when you don't need performance but just need capacity. If you have high performance needs, start looking at the SSD options. The persistent disks offer data redundancy because the data on each persistent disk is distributed across several physical disks.

Local SSDs provide even higher performance, but without the data redundancy.

Finally, RAM disks are very volatile but they provide the highest performance.

Maximum persistent disks

Machine type	Disk number limit
Shared-core	16
Standard	128
High-memory	
High-CPU	
Memory-optimized	
Compute-optimized	

Now, just as there is a limit on how many Local SSDs you can attach to a VM, there is also a limit on how many persistent disks you can attach to a VM. As illustrated in this table, this limit depends on the machine type. For the Shared-core machine type, you can attach up to 16 disks. For the Standard, High Memory, High-CPU, Memory-optimized, and Compute-optimized machine types, you can attach up to 128 disks. So you can create massive amounts of capacity for a single host.

Now remember that little nuance when I told you about how throughput is limited by the number of cores that you have? That throughput also shares the same bandwidth with Disk IO. So if you plan on having a large amount of Disk IO throughput, it will also compete with any network egress or ingress throughput. So remember that, especially if you will be increasing the number of drives attached to a virtual machine.

Persistent disk management differences

Cloud Persistent Disk	Computer Hardware Disk
	
• Single file system is best • Resize (grow) disks • Resize file system • Built-in snapshot service • Automatic encryption	• Partitioning • Repartition disk • Reformat • Redundant disk arrays • Subvolume management and snapshots • Encrypt files before write to disk

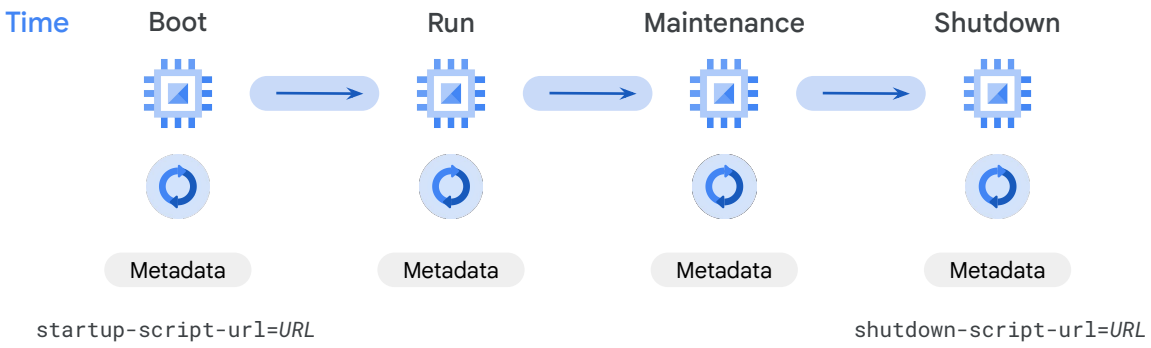
There are many differences between a physical hard disk in a computer and a persistent disk, which is essentially a virtual networked device. First of all, if you remember with normal computer hardware disks, you have to partition them. Essentially, you have a drive and you're carving up a section for the operating system to get its own capacity. If you want to grow it, you have to repartition, and if you want to make changes you might even have to reformat. If you want redundancy, you might create a redundant disk array, and if you want encryption, you need to encrypt files before writing them to the disk.

With cloud persistent disks, things are very different because all that management is handled for you on the backend. You can simply grow disks and resize the file system because disks are virtual networked devices. Redundancy and snapshot services are built in and disks are automatically encrypted. You can even use your own keys, and that will ensure that no party can get to the data except you.

Common Compute Engine Actions

Now that we have covered all the different compute, image, and disk options, let's look at some common actions that you can perform with Compute Engine.

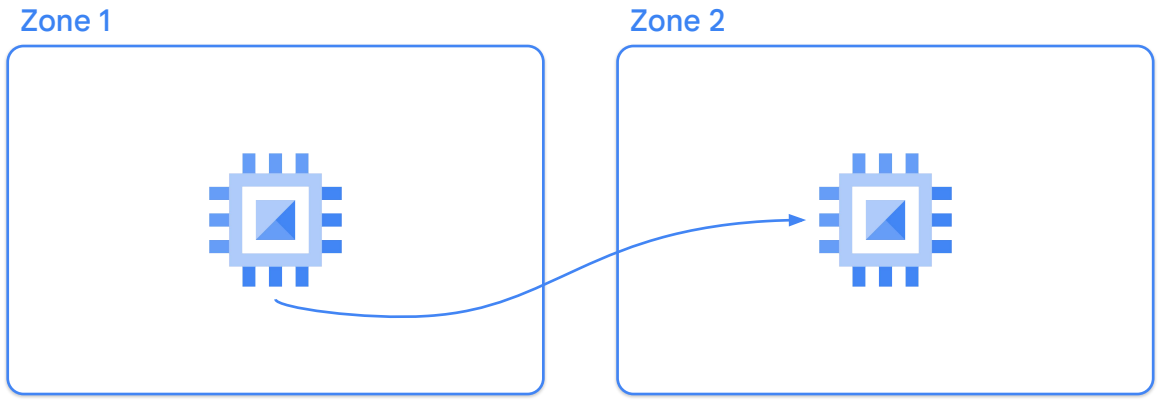
Metadata and scripts



Every VM instance stores its metadata on a metadata server. The metadata server is particularly useful in combination with startup and shutdown scripts, because you can use the metadata server to programmatically get unique information about an instance, without additional authorization. For example, you can write a startup script that gets the metadata key/value pair for an instance's external IP address and use that IP address in your script to set up a database. Because the default metadata keys are the same on every instance, you can reuse your script without having to update it for each instance. This helps you create less brittle code for your applications.

Storing and retrieving instance metadata is a very common Compute Engine action. We recommend storing the startup and shutdown scripts in Cloud Storage, as you will explore in the upcoming lab of this module.

Move an instance to a new zone



`gcloud compute instances move`

Another common action is to move an instance to a new zone. For example, you might do so for geographical reasons or because a zone is being deprecated.

You can move a VM even if one of the following scenarios applies:

- The VM instance is in a TERMINATED state.
- The VM instance is a Shielded VM that uses UEFI firmware.

Move an instance to a new zone

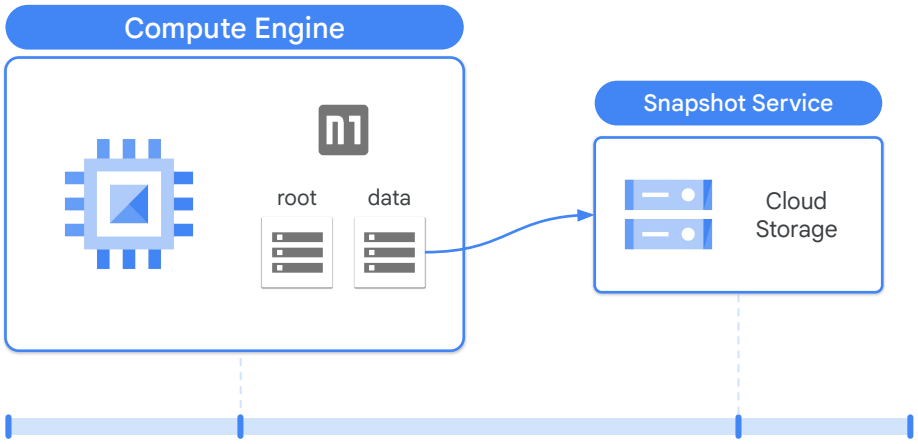
- **Automated process** (moving within region):
  - `gcloud compute instances move`
  - Update references to VM; not automatic
- **Manual process** (moving between regions):
  - Snapshot all persistent disks on the source VM.
  - Create new persistent disks in destination zone restored from snapshots.
  - Create new VM in the destination zone and attach new persistent disks.
  - Assign static IP to new VM.
  - Update references to VM.
  - Delete the snapshots, original disks, and original VM.

If you move your instance within the same region, you can automate the move by using the gcloud compute instances move command. To move your VM, you must shut down the VM, move it to the destination zone or region, and then restart it. After you move your VM, update any references that you have to the original resource, such as any target VMs or target pools that point to the earlier VM. During the move, some server-generated properties of your VM and disks change.

If you move your instance to a different region, you need to manually do so by following the process outlined here. This involves making a snapshot of all persistent disks and creating new disks in the destination zone from that snapshot. Next, you create the new VM in the destination zone and attach the new persistent disks, assign a static IP, and update any references to the VM. Finally, you delete the original VM, its disks, and the snapshot.

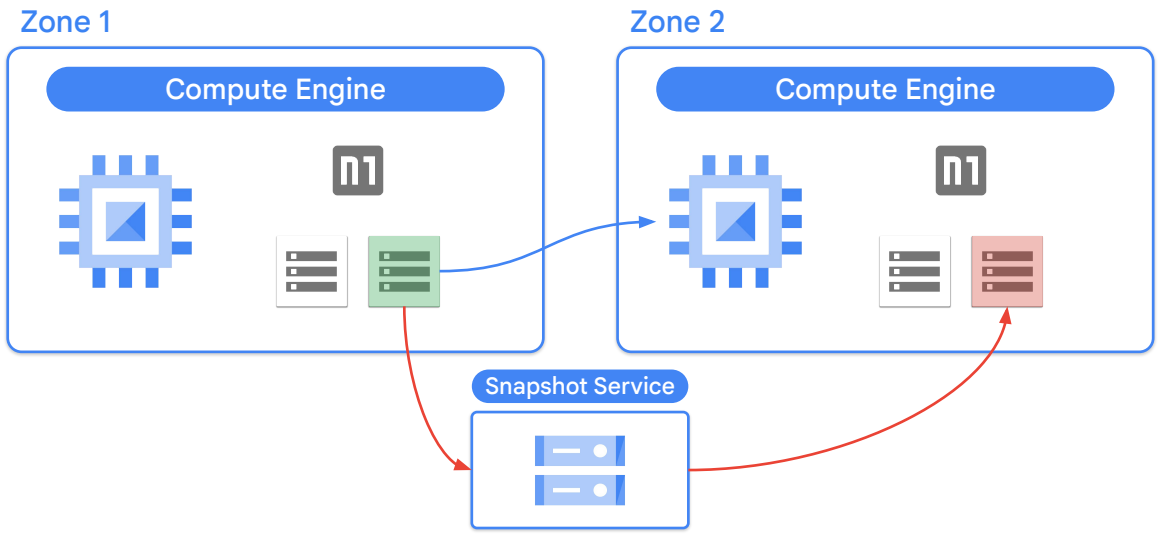
Speaking of snapshots, let’s take a closer look at these.

Snapshot: Backup critical data



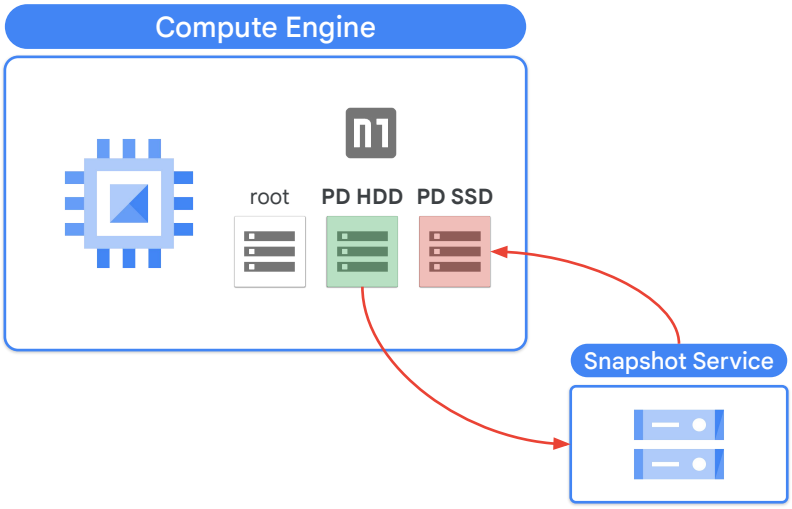
Snapshots have many use cases. For example, they can be used to backup critical data into a durable storage solution to meet application, availability, and recovery requirements. These snapshots are stored in Cloud Storage, which is covered later.

Snapshot: Migrate data between zones



Snapshots can also be used to migrate data between zones. We just discussed this when going over the manual process of moving an instance between two regions, but this can also be used to simply transfer data from one zone to another. For example, you might want to minimize latency by migrating data to a drive that can be locally attached in the zone where it is used.

Snapshot: Transfer to SSD to improve performance



Which brings me to another snapshot use case of transferring data to a different disk type. For example, if you want to improve disk performance, you could use a snapshot to transfer data from a standard HDD persistent disk to a SSD persistent disk.

Persistent disk snapshots

- Snapshot is not available for local SSD.
- Creates an ***incremental*** backup to Cloud Storage.
  - Not visible in your buckets; managed by the snapshot service.
- Create scheduled snapshots.
  - Regularly and automatically back up your zonal and regional persistent disks.
- Snapshots can be restored to a new persistent disk.
  - New disk can be in another region or zone in the same project.
  - Basis of VM migration: "moving" a VM to a new zone.
    - Snapshot doesn't backup VM metadata, tags, etc.

Now that we've covered some of the snapshot use cases, let's explore the concept of a disk snapshot.

First of all, this slide is titled persistent disk snapshots because snapshots are available only to persistent disks and not to local SSDs.
Snapshots are different from public images and custom images, which are used primarily to create instances or configure instance templates, in that snapshots are useful for periodic backup of the data on your persistent disks.

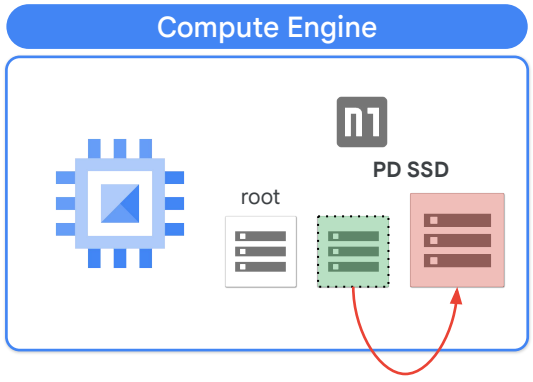
Snapshots are incremental and automatically compressed, so you can create regular snapshots on a persistent disk faster and at a much lower cost than if you regularly created a full image of the disk.

You can create a snapshot schedule to regularly and automatically back up your zonal and regional persistent disks.

As we saw with the previous examples, snapshots can be restored to a new persistent disk, allowing for a move to a new zone.

To create a persistent disk snapshot, refer to the module PDF under Course Resources. [https://cloud.google.com/compute/docs/disks/create-snapshots]

Resize persistent disk



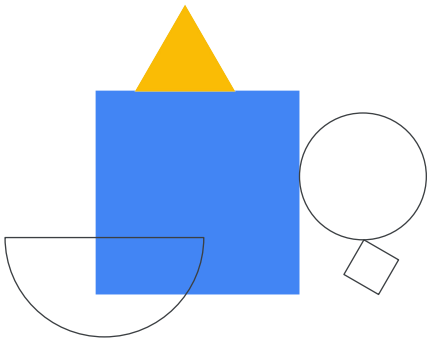
You can grow disks, but never shrink them!

Another common Compute Engine action is to resize your persistent disk. The added benefit of increasing storage capacity is to improve I/O performance. This can be achieved while the disk is attached to a running VM without having to create a snapshot.

Now, while you can grow disks in size, you can never shrink them, so keep this in mind.

Lab Intro

Working with Virtual Machines



Let’s get started with the second lab of this module.

In this lab, you'll be setting up an application server. Now this example happens to be a gaming application, but it applies to many other use cases. You will configure the VM and also add capacity for a production gaming system, and you will build the infrastructure that you need for production activities. This includes backups and graceful shutdown and restart services.

Lab Review

Working with Virtual Machines

In this lab, you created a customized virtual machine instance by installing base software, which was a headless Java runtime environment and application software, specifically, a Minecraft game server.

You customized the VM by preparing and attaching a high-speed SSD, and you reserved a static external IP address so that the address would remain consistent.

Using that IP address, you then verified the availability of the gaming server online.

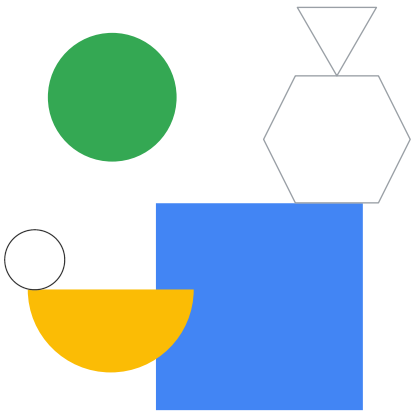
Next, you set up a backup system to back up the server's data to a Cloud Storage bucket, and you then tested that backup system. You then automated backups using cron.

Finally, you set up maintenance scripts using metadata for graceful startup and shutdown of the server.

Many of these techniques, including the script automation, can be adapted to administration of production servers in any application.

You can stay for a lab walkthrough, but remember that Google Cloud's user interface can change, so your environment might look slightly different.

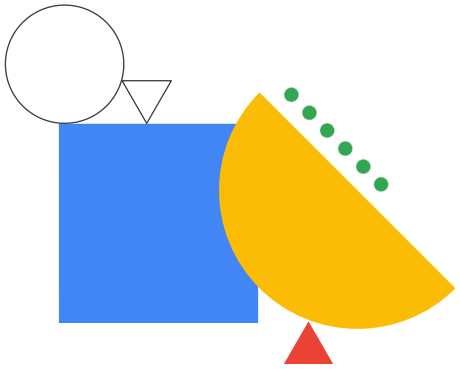
Review: Virtual Machines



In this module, we covered the different compute, image, and disk options within Compute Engine, along with some common actions. The two labs provided you with real world applications of most of the topics covered in this course.

Remember that there are many compute options to choose from. If a predefined machine type does not meet your needs, you can customize your own VM and you can even create a sole-tenant node. You can also install different public and custom images on the boot disks of your instances and you can attach more disks if needed.

Review: Essential Cloud Infrastructure: Foundation



Thank you for taking the Essential Cloud Infrastructure: Foundation course. I hope you have a better understanding of how to architect with Compute Engine, and I also hope that the demos and labs made you feel more comfortable with using the different GCP services that we covered.

Essential Cloud Infrastructure: Core Services

01 Cloud IAM

02 Data Storage Services

03 Resource Management

04 Resource Monitoring

Next, I recommend enrolling in the “Essential Cloud Infrastructure: Core Services” course of the “Architecting with Google Compute Engine” series.

In that course, we start by talking about Cloud IAM, and you will administer Identity and Access Management for resources.

Next, we’ll cover the different data storage services in GCP, and you will implement some of those services.

Then, we’ll go over resource management, where you will manage and examine billing of GCP resources.

Lastly, we’ll talk about resource monitoring, and you will monitor GCP resources using Stackdriver services.

Enjoy that course!