

Google Cloud Platform (GCP): 2.2

Planejamento e configuração de recursos de computação.



Prashanta Paudel

16 de outubro de 2018 · 8 minutos de leitura



Um dos aspectos mais importantes do GCP são os recursos de computação. Neste blog, veremos como fazemos o planejamento e a configuração nos recursos de computação.

Já deve estar claro que ***não planejamos um recurso computacional sozinho, mas planejamos um sistema ou serviços que precisarão de recursos computacionais baseados na plataforma google cloud.***

Como usar o Google Cloud Platform | Galeria de soluções | Google Cloud

Pesquise a documentação do GCP para tutoriais e soluções.

cloud.google.com

Este site lista várias soluções e qual infraestrutura é usada para implementar esse sistema.


Podemos escolher entre uma variedade de opções para usar para o problema específico em questão. Às vezes, é possível que o mesmo

problema possa ser resolvido usando 10 métodos diferentes.


O uso da opção específica de google computing depende de sua necessidade e dos recursos disponíveis nesse produto.

O Google mencionou algumas das necessidades e recursos e casos de uso comuns.


As capturas de tela foram tiradas do site do Google Cloud.

Product	Your needs	Product features	Typical use cases
 <p>Google Compute Engine</p> <p>Virtual machines running in Google's global data center network</p>	<ul style="list-style-type: none"> • You need complete control over your infrastructure and direct access to high-performance hardware such as GPUs and local SSDs. • You need to make OS-level changes, such as providing your own network or graphic drivers, to squeeze out the last drop of performance. • You want to move your application from your own colo or datacenter to the cloud without rewriting it. • You need to run a software package that can't easily be containerized or you want to use existing VM images. 	<ul style="list-style-type: none"> • Virtual machines with network-attached and ultra-high performance local storage options. • Preemptible virtual machines for inexpensive batch jobs and fault-tolerant workloads. • Customizable load-balancing and auto-scaling across homogeneous VMs. • Direct access to GPUs that you can use to accelerate specific workloads. • Support for the most popular flavors of Linux and Windows operating systems. 	<ul style="list-style-type: none"> • Any workload requiring a specific OS or OS configuration. • Currently deployed, on-premises software that you want to run in the cloud.

Casos de uso do Compute Engine

Product	Your needs	Product features	Typical use cases
 <p>Google Kubernetes Engine</p> <p>Logical infrastructure powered by Kubernetes, the open source container orchestration system.</p>	<ul style="list-style-type: none"> • You want to increase velocity and improve operability dramatically by separating the app from the OS. • You need a secure, scalable way to manage containers in production. • You don't have dependencies on a specific operating system. 	<ul style="list-style-type: none"> • Logical infrastructure - focus on your app components, not virtual machines. • Easy mechanisms for building loosely-coupled distributed systems. • Run the same application on your laptop, on premise and in the cloud. 	<ul style="list-style-type: none"> • Containerized workloads. • Cloud-native distributed systems. • Hybrid applications.

Casos de uso do Kubernetes

Product	Your needs	Product features	Typical use cases
 <p>Google App Engine</p> <p>A flexible, zero ops platform for building highly available apps</p>	<ul style="list-style-type: none"> You want to focus on writing code, and never want to touch a server, cluster, or infrastructure. You want to build a highly reliable and scalable serving app or component without doing it all yourself. You value developer velocity over infrastructure control. You want to minimize operational overhead. 	<ul style="list-style-type: none"> A range of curated serving stacks with smart defaults and deep customizability. Support for Java, Python, PHP, Go, Ruby, Node.js, and ASP.NET Core (beta) ... or bring your own app runtime. Integrated SDK, managed services, and local development environment. App versioning with zero-downtime upgrades. Traffic splitting. Automatic high availability with built-in auto-scaling. 	<ul style="list-style-type: none"> Web sites. Mobile app and gaming backends. RESTful APIs. Internal Line of Business (LOB) apps. Internet of things (IoT) apps.

Casos de uso do App Engine

Google Compute Engine (GCE)



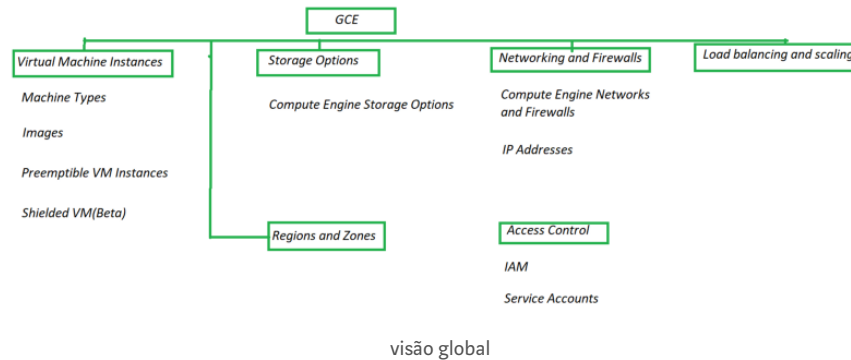
O mecanismo de computação do Google permite que os usuários criem e executem máquinas virtuais no Google Cloud Platform. Ele oferece escala, desempenho e valor que nos permitem lançar facilmente um grande cluster de computação na infraestrutura do Google. Usar o mecanismo de computação não requer nenhum custo inicial. O ferramental e o suporte ao fluxo de trabalho do Compute Engine permitem o dimensionamento de instâncias únicas para computação em nuvem global com balanceamento de carga.

A VM do mecanismo de computação vem com vários tipos de discos, incluindo disco permanente e SSD. Você também pode criar uma VM personalizada com a quantidade de disco e memória escolhida. Além

disso, você receberá um desconto se for executado por um longo período de tempo.

Você pode criar um grande cluster de recursos de computação e conectá-los a outros data centers com uma rede rápida e eficiente do google.

O Google Compute Engine consiste em seguir



Portanto, é óbvio que você precisa ter muito cuidado ao selecionar qualquer solução com base nos elementos do mecanismo de computação.

Instâncias de Máquina Virtual

Uma *instância* é uma máquina virtual (VM) hospedada na infraestrutura do Google. Você pode criar uma instância usando o Console do Google Cloud Platform ou a ferramenta de linha de comando.

Introdução

As instâncias de mecanismo de computação podem executar as imagens públicas para o Linux e as janelas que o Google fornece explicitamente em suas plataformas, além de poder criar ou importar de seus sistemas existentes. Você também pode implantar contêineres do Docker, que são iniciados automaticamente em instâncias que executam a imagem pública do Container-Optimized OS. Durante a criação, você pode selecionar não CPUs virtuais e memória usando um conjunto de tipos de máquinas predefinidos ou criando seu próprio tipo de máquina personalizado.

Instâncias e projetos

As instâncias pertencem ao projeto de console do GCP e um projeto pode ter várias instâncias. Você precisa definir zona, sistema operacional e tipo de máquina ao criar instâncias. Quando você exclui a instância, ela é removida do projeto.

Instâncias e opção de armazenamento

Cada instância possui um disco permanente no qual o sistema operacional está instalado. Se necessário, outro disco pode ser anexado à instância.

Instâncias e redes

Cada instância de computação, quando criada, tem uma VPC (nuvem privada virtual). **Até 5 VPC podem ser adicionados ao projeto.** Instâncias no mesmo VPC se comunicam na rede de área local. O IP público é usado somente quando precisa se comunicar fora do projeto.

Instâncias e contêineres

As instâncias do Compute Engine dão suporte a um método declarativo para iniciar seus aplicativos usando contêineres. Ao criar uma VM ou um modelo de instância, você pode fornecer um nome de imagem do Docker e iniciar a configuração. O Compute Engine cuidará do restante, incluindo o fornecimento de uma imagem atualizada do Container-Optimized OS com o Docker instalado e o lançamento do contêiner quando a VM for inicializada.

Google diz:

Gerenciando o acesso a suas instâncias

Você pode gerenciar o acesso às suas instâncias usando um dos seguintes métodos:

- Instâncias do Linux:

1. Gerenciando o Acesso à Instância Usando o Login do SO, que permite associar chaves SSH à sua Conta do Google ou conta do G Suite e gerenciar o acesso administrativo ou não administrativo à instância por meio de funções do IAM. Se você se conectar a suas instâncias usando a ferramenta de linha de comando ou o SSH no console, o Compute

Engine poderá gerar automaticamente chaves SSH para você e aplicá-las à sua conta do Google ou à sua conta do G Suite.

2. Gerencie suas chaves SSH em metadados de projeto ou de instância, o que concede acesso de administrador a instâncias com acesso a metadados que não usam o Login do SO. Se você se conectar a suas instâncias usando a ferramenta de linha de comando ou o SSH no console, o Compute Engine poderá gerar automaticamente chaves SSH para você e aplicá-las aos metadados do projeto.

- Em instâncias do Windows Server:

Crie uma senha para uma instância do Windows Server

Tipos de máquinas

Ao implementar máquinas virtuais, você tem três opções

1. Faça o upload de uma VM personalizada do seu sistema.
2. Use as VMs padrão disponíveis no sistema
3. Construa uma VM personalizada a partir das opções disponíveis

Normalmente, definir uma VM inclui a seleção do tamanho da memória, CPUs virtuais, espaço em disco, etc.

Então, vamos ver as diferentes opções

Tipos de máquina predefinidos

Os tipos de máquina predefinidos possuem um conjunto fixo de recursos. Eles são gerenciados pelo Google Compute Engine e vêm em quatro classes

Tipos de máquina 1.Standard

Os tipos de máquinas padrão são adequados para tarefas que têm um equilíbrio entre as necessidades de CPU e memória. Os tipos de máquinas padrão possuem 3,75 GB de memória por vCPU.

Você pode ter 1 a 96 vCPUs e 3,75 a 360 GB de memória.

O uso de disco permanente é cobrado separadamente do preço do tipo de máquina.

Tipos da máquina 2.High-memory

Os tipos de máquinas de alta memória são ideais para tarefas que exigem mais memória em relação às vCPUs. Os tipos de máquinas de alta memória possuem 6,5 GB de memória do sistema por vCPU.

Você pode ter 2 a 96 vCPUs e 13 a 624 GB de memória.

3. Tipos de máquinas com alta CPU

Os tipos de máquinas com alta CPU são ideais para tarefas que exigem mais vCPUs em relação à memória. Os tipos de máquinas com alta CPU têm 0,90 GB de memória por vCPU.

Você pode ter 2 a 96 vCPUs e 1,80 a 86,4 GB de memória

4. Tipos de máquinas com núcleo compartilhado

Os tipos de máquina de núcleo compartilhado fornecem uma vCPU que pode ser executada durante uma parte do tempo em um único hyperthread de hardware na CPU do host que está executando sua instância. As instâncias de núcleo compartilhado podem ser mais eficientes em termos de custo para executar aplicativos pequenos, sem uso intensivo de recursos, do que os tipos de máquina padrão, de alta memória ou de alta CPU.

f1-micro Estourando

Os tipos de máquinas f1-micro oferecem recursos de estouro que permitem que as instâncias usem CPU física adicional por curtos períodos de tempo. O estouro acontece automaticamente quando sua instância exige mais CPU física do que a alocada originalmente. Durante esses picos, sua instância aproveitará oportunisticamente a CPU física disponível em rajadas. Observe que as rajadas não são permanentes e só são possíveis periodicamente.

Machine name	Description	vCPUs	Memory (GB)	Max number of persistent disks (PDs) ¹	Max total PD size (TB)
f1-micro	Micro machine type with 0.2 vCPU, 0.60 GB of memory, backed by a shared physical core.	0.2	0.60	4 (16 in Beta)	3
g1-small	Shared-core machine type with 0.5 vCPU, 1.70 GB of memory, backed by a shared physical core.	0.5	1.70	4 (16 in Beta)	3

O uso de disco permanente é cobrado separadamente do preço do tipo de máquina.

5. Tipos de máquinas otimizadas para memória

Os tipos de máquinas otimizadas para memória são ideais para tarefas que exigem uso intensivo de memória com maior memória para proporções de vCPU do que para tipos de máquinas de alta memória. Esses tipos de máquinas são perfeitamente adequados para bancos de dados em memória e análises na memória, como cargas de trabalho de SAP Hana e Business Warehousing (BW), análise de genômica, serviços de análise SQL e muito mais. Os tipos de máquinas com memória otimizada têm mais de 14 GB de memória por vCPU.

Veja Regiões e Zonas para encontrar onde os tipos de máquinas com memória otimizada estão disponíveis.

Machine name	Description	vCPUs	Memory (GB)	Max number of persistent disks (PDs) ¹	Max total PD size (TB)	Local SSD
n1-ultramem-40	Memory-optimized machine type with 40 vCPUs and 961GB of memory.	40	961	16 (128 in Beta)	64	No
n1-ultramem-80	Memory-optimized machine type with 80 vCPUs and 1.87 TB of memory.	80	1922	16 (128 in Beta)	64	No
n1-megamem-96	Memory-optimized machine type with 96 vCPUs and 1.4 TB of memory.	96	1433.6	16 (128 in Beta)	64	Yes
n1-ultramem-160	Memory-optimized machine type with 160 vCPUs and 3.75 TB of memory.	160	3844	16 (128 in Beta)	64	No

Tipos de máquinas personalizadas

Se sua necessidade não corresponder a nenhum tipo de máquina predefinido, você poderá optar pelo tipo de máquina personalizado.

Os tipos de máquina personalizados são ideais para os seguintes cenários:

- Cargas de trabalho que não são adequadas para os tipos de máquinas predefinidos que estão disponíveis para você.
- Cargas de trabalho que exigem mais capacidade de processamento ou mais memória, mas não precisam de todas as atualizações fornecidas pelo próximo tipo de máquina predefinido maior.

Custa um pouco mais usar um tipo de máquina personalizado do que um tipo de máquina predefinido equivalente, e ainda há algumas limitações na quantidade de memória e vCPUs que você pode selecionar.

GPUs e tipos de máquinas

Você pode anexar GPUs apenas a instâncias com um tipo de máquina predefinido ou um tipo de máquina personalizado que você possa criar em uma zona. As GPUs não são suportadas em tipos de máquinas de núcleo compartilhado ou em tipos de máquinas com otimização de memória.

Instâncias preemptivas de VM

Essas instâncias que podem ser encerradas a qualquer momento pelo mecanismo de cálculo são chamadas de instâncias de VM preventivas. Essas VMs podem ser implementadas a um preço menor do que as instâncias normais, pois o mecanismo de computação tem o direito de eliminá-lo a qualquer momento.

Esses tipos de instâncias são melhores para aplicativos tolerantes a falhas, como o processamento em lote, que não afeta completamente o sistema, mesmo que esteja desativado ou continue mais tarde.

Instâncias preemptivas funcionam como instâncias normais, mas possuem as seguintes limitações:

- O Compute Engine pode encerrar as instâncias preemptivas a qualquer momento devido a eventos do sistema. A probabilidade de o Compute Engine encerrar uma instância preemptiva de um evento do sistema é geralmente baixa, mas pode variar de um dia para outro e de uma zona para outra dependendo das condições atuais.
- O Compute Engine sempre encerra as instâncias preemptivas depois de serem executadas por 24 horas.
- As instâncias preemptivas são recursos finitos do Compute Engine, portanto, podem nem sempre estar disponíveis.
- As instâncias preemptivas não podem ser migradas para uma instância regular da VM ou podem ser definidas para reiniciar automaticamente quando houver um evento de manutenção.
- Devido às limitações acima, as instâncias preemptivas não são cobertas por nenhum Contrato de Nível de Serviço (e, para maior clareza, são excluídas do SLA do Google Compute Engine).

Processo de preempção

O Compute Engine executa as seguintes etapas para antecipar uma instância:

1. O Compute Engine envia um aviso de preempção à instância na forma de um sinal ACPI G2 Soft Off. Você pode usar um script de desligamento para manipular o aviso de preempção e concluir as ações de limpeza antes que a instância pare.
2. Se a instância não parar após 30 segundos, o Compute Engine envia um sinal ACPI G3 Mechanical Off para o sistema operacional.
3. O Compute Engine transiciona a instância para um `TERMINATED` estado.

Você pode simular uma preempção de instância parando a instância.

