

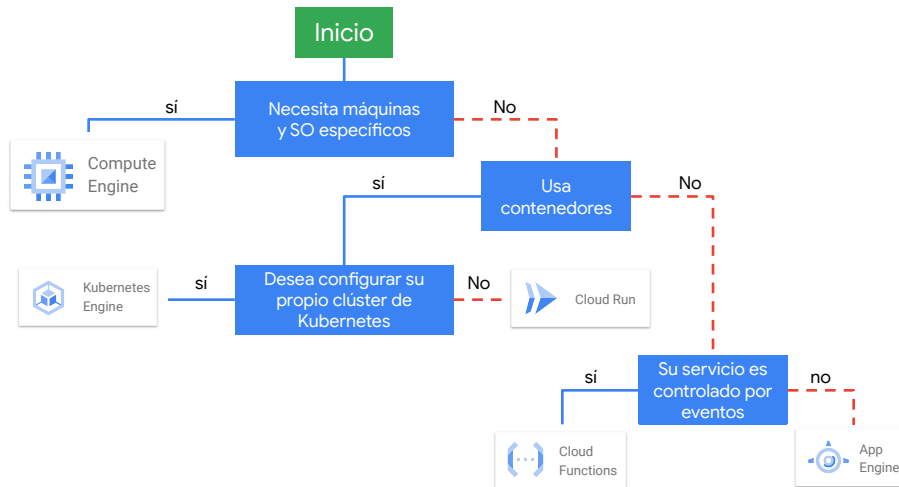


Cómo implementar aplicaciones en Google Cloud

Stephanie Wong
Developers Advocate, Google Cloud

En este módulo, analizaremos las distintas opciones para implementar aplicaciones en Google Cloud. Google Cloud cuenta con varias plataformas de implementación posibles, cuya elección no siempre es obvia.

Elija una plataforma de implementación de Google Cloud



Veamos una descripción general de cómo podría elegir la plataforma más adecuada para su aplicación.

Primero, pregúntese si necesita máquinas y SO específicos. En ese caso, elija Compute Engine.

Si no necesita máquinas y sistemas operativos específicos, la siguiente pregunta es si utiliza contenedores. En caso afirmativo, debería considerar Google Kubernetes Engine o Cloud Run, dependiendo de si desea configurar su propio clúster de Kubernetes.

Si no usa contenedores, le recomendamos considerar Cloud Functions si su servicio es controlado por eventos y App Engine si no lo es.

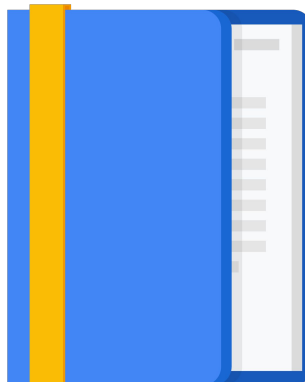
Explicaré cada uno de estos servicios en este módulo y podrá explorarlos en un lab.

Comencemos.

Temario

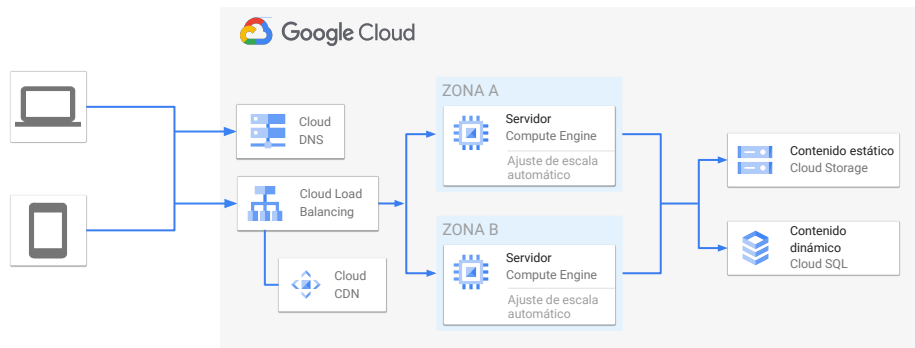
Infraestructura como servicio
de Google Cloud

Plataformas de implementación
de Google Cloud



Primero, hablemos de Compute Engine, la oferta de infraestructura como servicio (IaaS) de Google Cloud.

Use Compute Engine cuando necesite controlar totalmente los sistemas operativos o si tiene aplicaciones que no están alojadas en contenedores o que no son bases de datos autoalojadas



Compute Engine es una solución excelente cuando necesita controlar totalmente sus sistemas operativos o si tiene una aplicación que no está alojada en contenedores, una aplicación creada en una arquitectura de microservicios o una aplicación que es una base de datos.

Los grupos de instancias y el ajuste de escala automático, como se muestran en esta diapositiva, le permiten abordar las variaciones de demanda en su aplicación. Veamos con más detalle los grupos de instancias.

Los grupos de instancias administrados crean VMs basadas en plantillas de instancias

- Las plantillas de instancias definen las VMs: imágenes, tipo de máquina, etcétera.
 - Haga una prueba para encontrar el tipo de máquina más pequeño capaz de ejecutar su programa.
 - Use una secuencia de comandos de inicio para instalar su programa a partir de un repositorio de Git.
- El administrador del grupo de instancias crea las máquinas.
 - Configure el ajuste de escala automático para optimizar el costo y cumplir con las cargas de trabajo de usuarios variables.
 - Agregue una verificación de estado para permitir la reparación automática.
 - Use varias zonas para tener alta disponibilidad.

Los grupos de instancias administrados crean VMs basadas en plantillas de instancias. Las plantillas de instancias son solo un recurso que se usa para definir VMs y grupos de instancias administrados. Las plantillas definen la imagen de disco de arranque o de contenedor que se debe usar, el tipo de máquina, las etiquetas y otras propiedades de instancias como una secuencia de comandos de inicio para instalar software a partir de un repositorio de Git.

Un administrador de grupos de instancias crea las máquinas virtuales en un grupo de instancias administrado. Si usa un grupo de instancias administrado, tendrá muchas ventajas, como la reparación automática para volver a crear las instancias que no responden y crear instancias en varias zonas a fin de tener alta disponibilidad.

Use uno o más grupos de instancias como backends para los balanceadores de cargas

- Use un balanceador de cargas global si tiene grupos de instancias en varias regiones.
- Habilite la CDN para almacenar en caché el contenido estático.
- Para los servicios externos, configure SSL.
- Para los servicios internos, no proporcione una dirección IP pública.

Crear servicio de backend

Nombre ⓘ
El nombre es permanente
web-service-backend

Descripción ⓘ

Protocolo: HTTP Puerto con nombre: http Tiempo de espera: 30 segundos ⓘ

Tipo de backend
☒ Grupos de instancias
☐ Grupos de extremos de red

Backends
Regiones: europe-north1, us-central1

instance-group-europe (zona: europe-north1-a, puerto: 80) ⓘ
instance-group-us (zona: us-central1-a, puerto: 80) ⓘ

[+ Agregar backend](#)

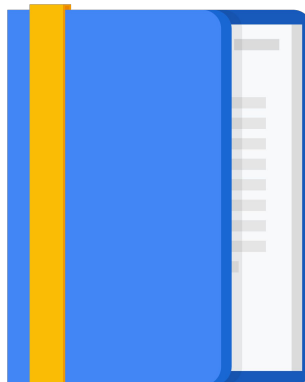
Cloud CDN ⓘ
☒ Habilitar Cloud CDN

Se recomienda usar uno o más grupos de instancias como backends para los balanceadores de cargas. Si necesita grupos de instancias en varias regiones, use un balanceador de cargas global. Si tiene contenido estático, solo habilite Cloud CDN como se muestra a la derecha.

Temario

Infraestructura como servicio
de Google Cloud

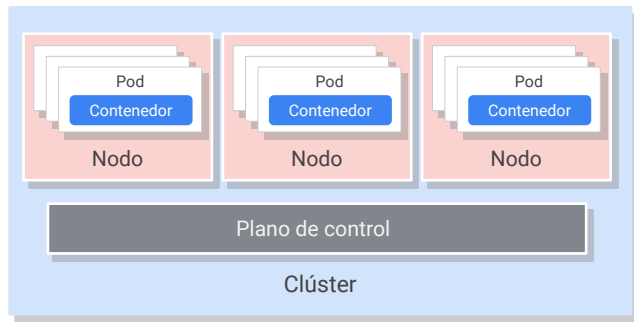
Plataformas de implementación
de Google Cloud



Veamos las otras plataformas de implementación: GKE, Cloud Run, App Engine y Cloud Functions.

Google Kubernetes Engine (GKE) automatiza la creación y administración de la infraestructura de procesamiento

- Los clústeres de Kubernetes tienen una colección de nodos.
- En GKE, los nodos son VMs de Compute Engine.
- Los servicios se implementan en Pods.
- Optimice el uso de recursos implementando varios servicios en el mismo clúster.
- Paga por las VMs.

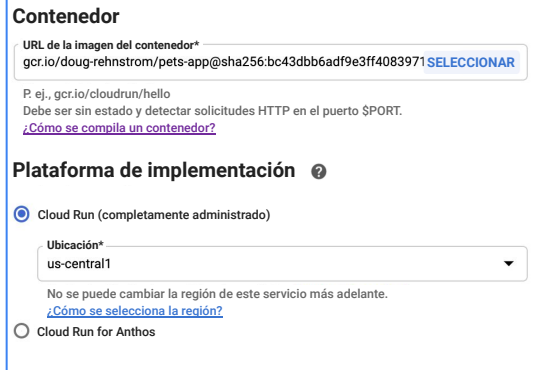


Google Kubernetes Engine (GKE) proporciona un entorno administrado para implementar, administrar y escalar aplicaciones alojadas en contenedores con la infraestructura de Google. El entorno de GKE consta de varias máquinas virtuales de Compute Engine unidas para formar un clúster. Los clústeres de GKE funcionan con el sistema de administración de clúster de código abierto de Kubernetes. Kubernetes otorga los mecanismos con los que interactuar con el clúster. Los comandos y recursos de Kubernetes se usan para implementar y administrar aplicaciones, realizar tareas de administración, establecer políticas y supervisar el estado de las cargas de trabajo implementadas.

En el diagrama de la derecha, se muestra el diseño de un clúster de Kubernetes. Un clúster consta de un plano de control del clúster como mínimo y varias máquinas trabajadoras denominadas nodos. Este plano de control y las máquinas de nodos ejecutan el sistema de organización de clústeres de Kubernetes. Los Pods son los objetos más pequeños y básicos que se pueden implementar en Kubernetes. Un Pod representa una instancia única de un proceso en ejecución en un clúster. Los Pods contienen un contenedor o más, como los contenedores de Docker, que ejecutan los servicios que se implementan. Puede optimizar el uso de recursos implementando varios servicios en el mismo clúster.

Cloud Run le permite implementar contenedores en clústeres de Kubernetes administrados por Google

- Cloud Run le permite usar Kubernetes sin administrar clústeres ni escribir código de configuración.
- Las aplicaciones deben ser sin estado.
- Se deben implementar aplicaciones con imágenes de Docker en Container Registry.
- También se puede usar Cloud Run para automatizar la implementación en su propio clúster de GKE.



Contenedor

URL de la imagen del contenedor*

`gcr.io/doug-rehnstrom/pets-app@sha256:bc43dbb6adf9e3ff4083971` [SELECCIONAR](#)

P. ej., `gcr.io/cloudrun/hello`
Debe ser sin estado y detectar solicitudes HTTP en el puerto \$PORT.
[¿Cómo se compila un contenedor?](#)

Plataforma de implementación ?

☒ Cloud Run (completamente administrado)

Ubicación*

`us-central1` ▼

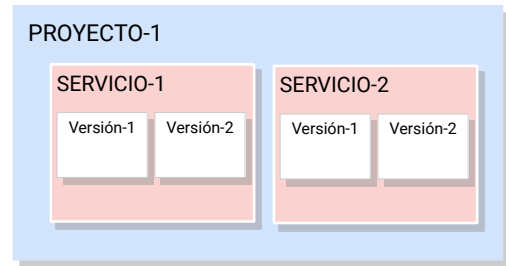
No se puede cambiar la región de este servicio más adelante.
[¿Cómo se selecciona la región?](#)

☐ Cloud Run for Anthos

Cloud Run, por otro lado, le permite implementar contenedores en un clúster de Kubernetes administrado por Google. Una gran ventaja es que no necesita administrar ni configurar el clúster. Los servicios que implementa deben ser sin estado y las imágenes que usa deben estar en Container Registry. Cloud Run se puede usar para automatizar la implementación en clústeres de GKE Anthos. Debe hacerlo si necesita más control de sus servicios, ya que le permitirá acceder a su red de VPC, ajustar el tamaño de las instancias de procesamiento y ejecutar sus servicios en todas las regiones de GKE. En la captura de pantalla de la derecha, se muestra una configuración de Cloud Run en la que la URL de la imagen de contenedor se especifica junto con la plataforma de implementación, que puede ser Cloud Run o Cloud Run for Anthos completamente administrada.

App Engine se diseñó para los microservicios

- Cada proyecto de Google Cloud puede contener 1 aplicación de App Engine.
- Una aplicación tiene 1 servicio o más.
- Cada servicio tiene 1 versión o más.
- Las versiones tienen 1 instancia o más.
- Cuenta con división del tráfico automática para las versiones que cambian.

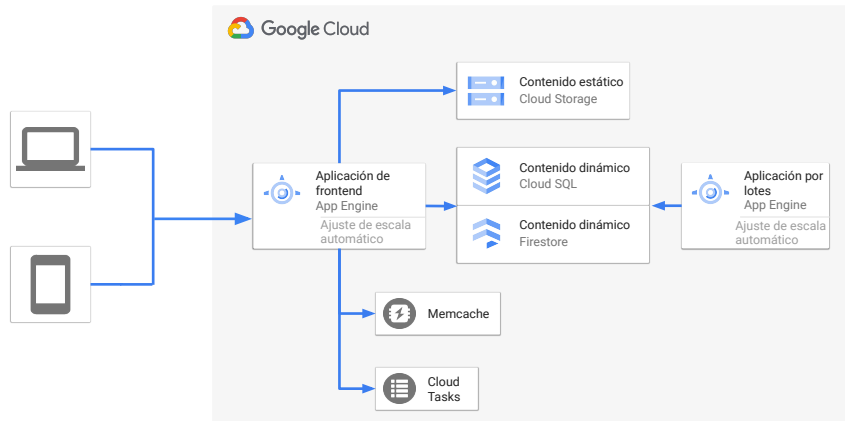


App Engine es una plataforma de aplicaciones completamente administrada y sin servidores que permite crear y administrar aplicaciones. Las aplicaciones se pueden escalar sin interrupciones desde cero sin tener que preocuparse de administrar la infraestructura subyacente. App Engine se diseñó para los microservicios. Para la configuración, cada proyecto de Google Cloud puede contener una aplicación de App Engine, que posee uno o más servicios. Cada servicio puede tener una o más versiones, que incluyen una o más instancias. App Engine admite la división del tráfico, por lo que es sencillo cambiar entre versiones y estrategias, como las pruebas de versiones canary o las pruebas A/B.

En el diagrama de la derecha, se muestra la organización de alto nivel de un proyecto de Google Cloud que tiene dos servicios, cada uno con dos versiones. La implementación y el control de versiones de estos servicios son independientes.

Veamos una arquitectura de microservicios común de App Engine.

Arquitectura de microservicios común de App Engine

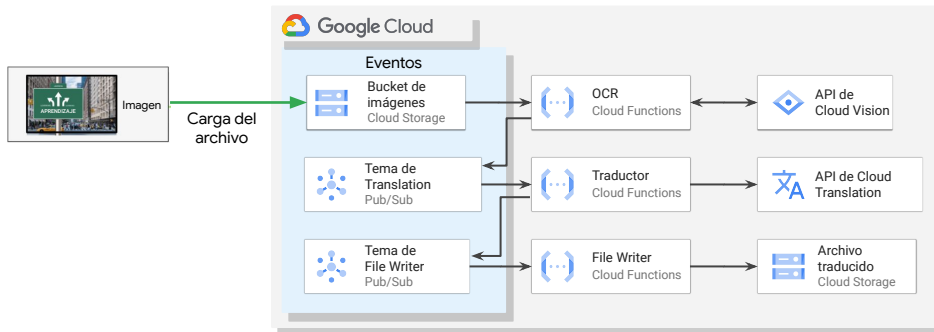


Este podría ser un ejemplo de un minorista que realiza ventas en línea. Aquí, App Engine funciona como el frontend para los clientes de la Web y de dispositivos móviles. El backend de esta aplicación es una variedad de soluciones de almacenamiento de Google Cloud con contenido estático (como imágenes almacenadas en Cloud Storage), Cloud SQL usado para datos relacionales estructurados (como datos del cliente y de ventas) y Firestore usado para el almacenamiento NoSQL (como datos del producto). Firestore tiene el beneficio de poder sincronizarse con aplicaciones de clientes.

Memcache se usa para disminuir la carga de los almacenes de datos guardando en caché las consultas. Cloud Tasks se utiliza a fin de realizar trabajos de forma asíncrona fuera de la solicitud de un usuario (o solicitud entre servicios). También hay una aplicación por lotes que genera informes de datos para la administración.

Cloud Functions es una forma excelente de crear microservicios con acoplamiento bajo y controlados por eventos

- Se puede activar mediante cambios realizados en un bucket de almacenamiento, mensajes de Pub/Sub o solicitudes web.
- Es completamente administrado, escalable y de bajo costo.



Cloud Functions es una forma excelente de implementar microservicios con acoplamiento bajo y controlados por eventos. Se diseñaron para los eventos de procesamiento que ocurren en Google Cloud. Las funciones se pueden activar mediante cambios realizados en un bucket de Cloud Storage, un mensaje Pub/Sub o solicitudes HTTP. La plataforma es completamente administrada, escalable y de bajo costo. No paga si no hay solicitudes y el procesamiento se paga por tiempo de ejecución en incrementos de 100 ms.

En el gráfico, se ilustra un servicio de traducción de imágenes implementado con Cloud Functions. Cuando se sube una imagen a un bucket de Cloud Storage, se activa una Cloud Function de OCR capaz de identificar el texto de la imagen mediante la API de Cloud Vision de Google. Una vez que se identifica el texto, este servicio publica un mensaje en un tema de Pub/Sub para que se traduzca, lo que activa otra Cloud Function que traducirá el texto identificado en la imagen usando la API de Cloud Translation. Después, la Cloud Function traductora publicará un mensaje en un tema de escritura de archivo en Pub/Sub, lo cual activará una Cloud Function que escribirá la imagen traducida en un archivo.

En la secuencia, se ilustra un caso de uso común de Cloud Functions para el procesamiento basado en eventos.

Lab

Cómo implementar aplicaciones en Google Cloud



App Engine



Google
Kubernetes
Engine



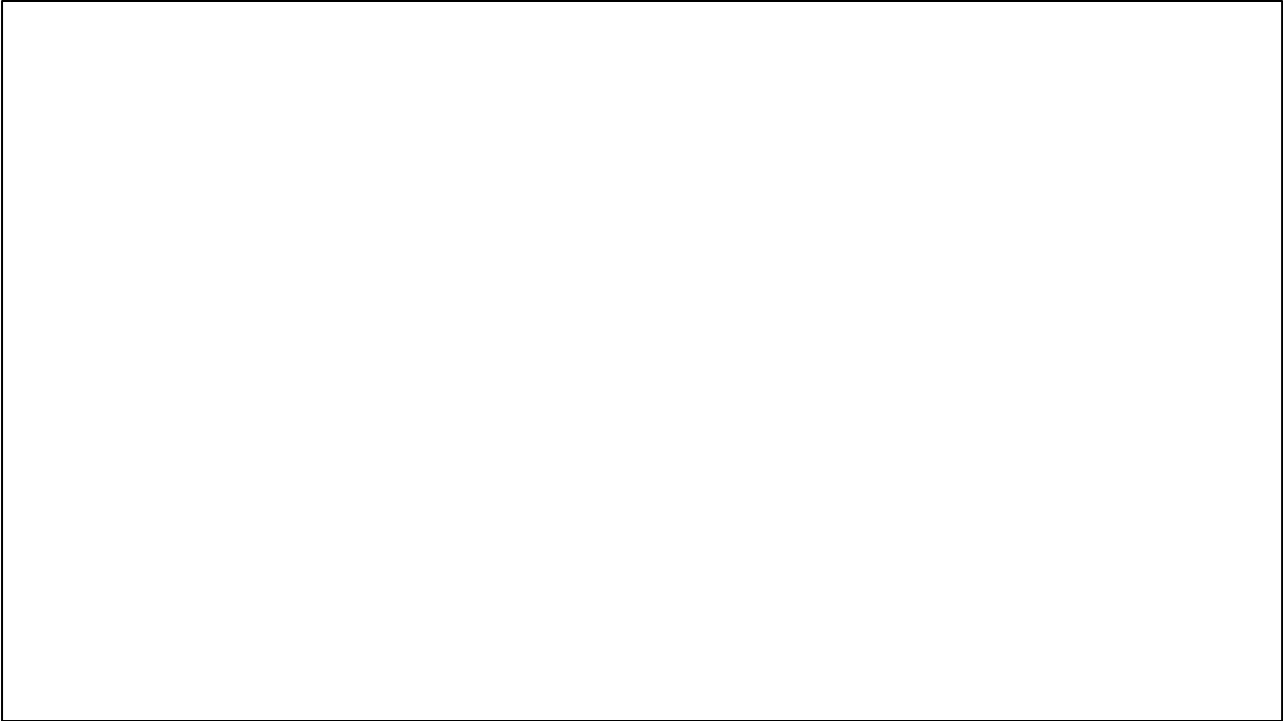
Cloud Run

Objetivos

- Realizar una implementación en App Engine
- Realizar una implementación en Google Kubernetes Engine
- Realizar una implementación en Cloud Run

En el primer lab de este curso, usó Cloud Build para crear imágenes de Docker y almacenarlas en Container Registry. Si sigue las prácticas recomendadas de 12 factores cuando escriba sus aplicaciones, debería poder crear aplicaciones portátiles en diferentes proveedores de servicios en la nube y también entre distintos servicios de implementación que otorga la nube. Por ello, en este lab implementará código en App Engine, Google Kubernetes Engine y Cloud Run. Veamos rápidamente por qué elegimos esos servicios para este lab:

Implementar su aplicación de Docker en una máquina virtual con Compute Engine en el primer lab fue sencillo, pero es posible que no sea la opción más eficaz. Una forma más automatizada podría ser usar App Engine, que será la primera plataforma de procesamiento de este lab. App Engine es excelente para quienes solo desean enfocarse en el código y no preocuparse para nada de la infraestructura subyacente (como las redes, los balanceadores de cargas y los escaladores automáticos), de cuya administración total se encarga App Engine.



Sin embargo, a veces los desarrolladores desean más libertad para personalizar sus entornos. Google Kubernetes Engine (GKE) otorga un equilibrio en donde tiene mucha personalización sobre su entorno de manera similar a Compute Engine. No obstante, GKE también lo ayuda a optimizar los gastos permitiéndole implementar varios servicios en el mismo clúster de máquinas virtuales, lo cual brinda un equilibrio excelente entre flexibilidad, portabilidad y optimización de costos.

Debe tener en cuenta que usar Kubernetes puede resultar complicado. Aquí es donde entra en juego Cloud Run, que le permite implementar sus propios servicios de Docker sin estado en clústeres de Kubernetes administrados por Google. Google Cloud se encarga de las partes difíciles de administrar el clúster y configurar los balanceadores de cargas, los escaladores automáticos y los verificadores de estado para que solo se tenga que enfocar en el código.

Es posible que implementar una sola aplicación en todas estas plataformas de procesamiento lo ayude a elegir la plataforma adecuada para sus propios servicios.

Repaso del lab

Cómo implementar
aplicaciones en
Google Cloud

En este lab, vio cómo implementar una aplicación en App Engine, GKE y Cloud Run. Esperamos que con este lab aprendiera cómo funciona cada una de estas plataformas de procesamiento y que le sirva a fin de elegir la plataforma adecuada para sus propios servicios.

Recomendamos que siga las prácticas recomendadas de 12 factores que vimos anteriormente en el curso, que automatice lo más posible con integración continua y herramientas de infraestructura como código y que aloje en contenedores sus servicios con Docker. Si lo hace, sus aplicaciones serán muy portátiles y la implementación será más sencilla y flexible, independientemente de la plataforma de procesamiento que use.

Puede continuar con un recorrido por el lab, pero recuerde que la interfaz de usuario de Google Cloud puede cambiar, por lo que su entorno podría ser un poco diferente.

Repaso

Cómo implementar aplicaciones en Google Cloud

En este módulo, abordamos los distintos servicios de implementación que ofrece Google. Estos incluyen Compute Engine si necesita tener el control completo de su entorno de implementación; Google Kubernetes Engine si desea la flexibilidad, portabilidad y automatización que otorga Kubernetes y App Engine, y Cloud Run si desea una plataforma como servicio completamente administrada.

Recuerde que todas estas opciones tienen ventajas y desventajas. Asegúrese de entenderlas todas para que pueda tomar una decisión fundamentada cuando implemente sus servicios.

Cuestionario

Debe implementar una aplicación existente escrita en .NET versión 4. La aplicación requiere servidores de Windows y no quiere cambiarla. ¿Qué plataforma debería usar?

- A. Compute Engine
- B. GKE
- C. App Engine
- D. Cloud Functions

Debe implementar una aplicación existente escrita en .NET versión 4. La aplicación requiere servidores de Windows y no quiere cambiarla. ¿Qué plataforma debería usar?

- A. Compute Engine
- B. GKE
- C. App Engine
- D. Cloud Functions

Cuestionario

Debe implementar una aplicación existente escrita en .NET versión 4. La aplicación requiere servidores de Windows y no quiere cambiarla. ¿Qué plataforma debería usar?

A. Compute Engine

B. GKE

C. App Engine

D. Cloud Functions

- A. Esta es la respuesta correcta. El enfoque es lift-and-shift, que es más compatible con Compute Engine, ya que ofrece un control completo de las máquinas virtuales, incluido el sistema operativo. No se necesitaría modificar ni reempaquetar la aplicación.
- B. Esta respuesta no es correcta. Para usar GKE, se necesitaría modificar y reempaquetar la aplicación en contenedores de Docker.
- C. Esta respuesta no es correcta. El entorno estándar de App Engine no es compatible con .NET. Además, para usar el entorno flexible de App Engine, se necesitaría modificar y reempaquetar la aplicación.
- D. Esta respuesta no es correcta. Cloud Functions es el modelo equivocado, ya que se diseñó para implementar funciones de propósito único.

A es la respuesta correcta. El enfoque es lift-and-shift, que es más compatible con Compute Engine, ya que ofrece un control completo de las máquinas virtuales, incluido el sistema operativo. No se necesitaría modificar ni reempaquetar la aplicación.

La respuesta B no es correcta. Para usar GKE, se necesitaría modificar y reempaquetar la aplicación en contenedores de Docker.

La respuesta C no es correcta. El entorno estándar de App Engine no es compatible con .NET. Además, para usar el entorno flexible de App Engine, se necesitaría modificar y reempaquetar la aplicación.

La respuesta D no es correcta. Cloud Functions es el modelo equivocado, ya que se diseñó para implementar funciones de propósito único.

Cuestionario

Alojó en contenedores varias aplicaciones con Docker y las implementó con VMs de Compute Engine. Desea ahorrar en costos y simplificar la administración del contenedor. ¿Qué puede hacer?

- A. Escribir secuencias de comandos de Terraform para todas las implementaciones
- B. Volver a escribir las aplicaciones que se deben ejecutar en App Engine
- C. Volver a escribir las aplicaciones que se deben ejecutar en Cloud Functions
- D. Migrar los contenedores a GKE

Alojó en contenedores varias aplicaciones con Docker y las implementó con VMs de Compute Engine. Desea ahorrar en costos y simplificar la administración del contenedor. ¿Qué puede hacer?

- A. Escribir secuencias de comandos de Terraform para todas las implementaciones
- B. Volver a escribir las aplicaciones que se deben ejecutar en App Engine
- C. Volver a escribir las aplicaciones que se deben ejecutar en Cloud Functions
- D. Migrar los contenedores a GKE

Cuestionario

Alojó en contenedores varias aplicaciones con Docker y las implementó con VMs de Compute Engine. Desea ahorrar en costos y simplificar la administración del contenedor. ¿Qué puede hacer?

- A. Escribir secuencias de comandos de Terraform para todas las implementaciones
- B. Volver a escribir las aplicaciones que se deben ejecutar en App Engine
- C. Volver a escribir las aplicaciones que se deben ejecutar en Cloud Functions
- D. Migrar los contenedores a GKE

- A. Esta respuesta no es correcta. Si bien esto se podría lograr con Terraform, no se cumpliría la necesidad de ahorrar en costos y simplificar la administración del contenedor.
- B. Si las aplicaciones están alojadas en contenedores, reescribirlas para ejecutarlas en App Engine no es rentable.
- C. Esta respuesta no es correcta. Cloud Functions se diseñó para implementar funciones de propósito único, no aplicaciones.
- D. Esta es la respuesta correcta. Las aplicaciones están alojadas en contenedores y GKE ayudará con la eficiencia de los recursos y, por lo tanto, con los costos. Esta solución automatizará muchos aspectos de la administración de los contenedores y es la mejor opción para este caso.

D es la respuesta correcta. Las aplicaciones están alojadas en contenedores y GKE ayudará con la eficiencia de los recursos y, por lo tanto, con los costos. Esta solución automatizará muchos aspectos de la administración de los contenedores y es la mejor opción para este caso.

La respuesta A no es correcta. Si bien esto se podría lograr con Terraform, no se cumpliría la necesidad de ahorrar en costos y simplificar la administración del contenedor.

La respuesta B no es correcta. Si las aplicaciones están alojadas en contenedores, reescribirlas para ejecutarlas en App Engine no es rentable.

La respuesta C no es correcta. Cloud Functions se diseñó para implementar funciones de propósito único, no aplicaciones.

Cuestionario

Se le solicitó escribir un programa que use la API de Vision para que verifique contenido inapropiado en fotos que se suban a un bucket de Cloud Storage. Se deben borrar todas las fotos que sean inapropiadas. ¿Cuál podría ser la forma más sencilla y económica de implementar este programa?

- A. Compute Engine
- B. GKE
- C. Cloud Functions
- D. App Engine

Se le solicitó escribir un programa que use la API de Vision para que verifique contenido inapropiado en fotos que se suban a un bucket de Cloud Storage. Se deben borrar todas las fotos que sean inapropiadas. ¿Cuál podría ser la forma más sencilla y económica de implementar este programa?

- A. Compute Engine
- B. GKE
- C. Cloud Functions
- D. App Engine

Cuestionario

Se le solicitó escribir un programa que use la API de Vision para que verifique contenido inapropiado en fotos que se suban a un bucket de Cloud Storage. Se deben borrar todas las fotos que sean inapropiadas. ¿Cuál podría ser la forma más sencilla y económica de implementar este programa?

A. Compute Engine

B. GKE

C. Cloud Functions

D. App Engine

C es la respuesta correcta. Con Cloud Functions, se cumplen los requisitos de sencillez y economía. Cloud Functions se diseñó para funciones de propósito único, como el análisis de imágenes. También se puede activar mediante eventos de Cloud Storage, por lo que otorga integración continua. El modelo de pago basado en la cantidad de solicitudes, el tiempo de procesamiento de las solicitudes (medido en unidades de 100 ms) y otros recursos consumidos es la opción más adecuada de todas las anteriores. Además, se ofrece un nivel gratuito. Cloud Functions también proporciona ajuste de escala automático, alta disponibilidad y tolerancia a errores.

Las respuestas A, B y D podrían ser soluciones, pero necesitarían más trabajo de desarrollo y más gastos para los recursos usados y, como resultado, no cumplirían con el requisito de sencillez y economía a fin de lograr la funcionalidad que se necesita.

C es la respuesta correcta. Con Cloud Functions, se cumplen los requisitos de sencillez y economía. Cloud Functions se diseñó para funciones de propósito único, como el análisis de imágenes. También se puede activar mediante eventos de Cloud Storage, por lo que otorga integración continua. El modelo de pago basado en la cantidad de solicitudes, el tiempo de procesamiento de las solicitudes (medido en unidades de 100 ms) y otros recursos consumidos es la opción más adecuada de todas las anteriores. Además, se ofrece un nivel gratuito. Cloud Functions también proporciona ajuste de escala automático, alta disponibilidad y tolerancia a errores.

Las respuestas A, B y D podrían ser soluciones, pero necesitarían más trabajo de desarrollo y más gastos para los recursos usados y, como resultado, no cumplirían con el requisito de sencillez y economía a fin de lograr la funcionalidad que se necesita.

Más recursos

Migración a Google Cloud: Implementa tus cargas de trabajo

<https://cloud.google.com/solutions/migration-to-gcp-deploying-your-workloads>

Compute Engine

<https://cloud.google.com/compute/>

GKE

<https://cloud.google.com/kubernetes-engine/>

App Engine

<https://cloud.google.com/appengine/>

Los vínculos otorgan acceso a algunos recursos útiles sobre las plataformas de implementación de Google Cloud.