Cómo crear implementaciones de Google Kubernetes Engine

1 hora5 créditos

Calificar lab

**Descripción general**

En este lab, explorarás los conceptos básicos del uso de manifiestos de implementación. Los manifiestos son archivos que contienen las configuraciones necesarias para una implementación que puede usarse en diferentes Pods. Los manifiestos son fáciles de cambiar.

**Objetivos**

En este lab, aprenderás a realizar las siguientes tareas:

* Crear manifiestos de implementación, realizar implementaciones en el clúster y verificar la reprogramación de Pods cuando se inhabilitan los nodos
* Activar el aumento y la reducción manual del escalamiento vertical de Pods en las implementaciones
* Activar el lanzamiento (actualización progresiva a la versión nueva) y las reversiones de la implementación
* Realizar una implementación de versiones canary

**Configuración del lab**

**Accede al lab**

En cada lab, recibirá un proyecto de Google Cloud y un conjunto de recursos nuevos por tiempo limitado y sin costo adicional.

1. Accede a Qwiklabs desde una **ventana de incógnito**.
2. Ten en cuenta el tiempo de acceso del lab (por ejemplo, 1:15:00) y asegúrate de finalizarlo en el plazo asignado.  
   No existe una función de pausa. Si lo necesita, puede reiniciar el lab, pero deberá hacerlo desde el comienzo.
3. Cuando esté listo, haga clic en **Comenzar lab**.
4. Anote las credenciales del lab (el **nombre de usuario** y la **contraseña**). Las usarás para acceder a la consola de Google Cloud.
5. Haga clic en **Abrir Google Console**.
6. Haga clic en **Usar otra cuenta**, copie las credenciales para **este** lab y péguelas en el mensaje emergente que aparece.  
   Si usa otras credenciales, se generarán errores o **incurrirá en cargos**.
7. Acepta las condiciones y omite la página de recursos de recuperación.

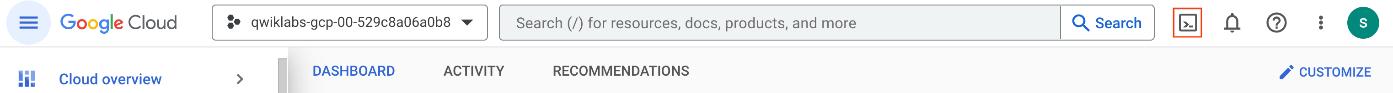
**Nota:** No hagas clic en **Finalizar lab**, a menos que lo hayas terminado o desees reiniciarlo. Esta acción borrará tu trabajo y quitará el proyecto.

Después de completar los pasos iniciales de acceso, aparecerá el panel del proyecto.

Active Google Cloud Shell

Google Cloud Shell es una máquina virtual que cuenta con herramientas de desarrollo. Ofrece un directorio principal persistente de 5 GB y se ejecuta en Google Cloud. Además, Google Cloud Shell proporciona acceso de línea de comandos a sus recursos de GCP.

1. En GCP Console, en la barra de herramientas superior derecha, haga clic en el botón Abrir Cloud Shell.



1. Haga clic en **Continue** (Continuar):



Puede hacer clic en INICIAR CLOUD SHELL de inmediato cuando se abra el cuadro de diálogo.

El aprovisionamiento y la conexión al entorno demoran solo unos minutos. Cuando esté conectado, ya estará autenticado y el proyecto estará configurado con su *PROJECT\_ID*. Por ejemplo:



**gcloud** es la herramienta de línea de comandos para Google Cloud Platform. Viene preinstalada en Cloud Shell y es compatible con la función de autocompletado con tabulador.

Puede solicitar el nombre de la cuenta activa con este comando:

gcloud auth list

content\_copy

Resultado:

Credentialed accounts:

- <myaccount>@<mydomain>.com (active)content\_copy

Resultado de ejemplo:

Credentialed accounts:

- google1623327\_student@qwiklabs.netcontent\_copy

Puede solicitar el ID del proyecto con este comando:

gcloud config list project

content\_copy

Resultado:

[core]

project = <project\_ID>content\_copy

Resultado de ejemplo:

[core]

project = qwiklabs-gcp-44776a13dea667a6content\_copy

La documentación completa de **gcloud** está disponible en [Descripción general de gcloud en Google Cloud](https://cloud.google.com/sdk/gcloud).

**Tarea 1. Crea manifiestos de implementación y realiza implementaciones en el clúster**

En esta tarea, crearás un manifiesto de implementación para un Pod dentro del clúster.

**Conéctate al clúster de GKE del lab**

1. En Cloud Shell, escribe el siguiente comando para establecer la variable de entorno para la zona y el nombre del clúster:

export my\_zone=us-central1-a

export my\_cluster=standard-cluster-1

Se copió correctamente

content\_copy

1. Configura el autocompletado con tabulador de kubectl en Cloud Shell:

source <(kubectl completion bash)

Se copió correctamente

content\_copy

1. En Cloud Shell, configura el acceso a tu clúster para la herramienta de línea de comandos de kubectl con el siguiente comando:

gcloud container clusters get-credentials $my\_cluster --zone $my\_zone

Se copió correctamente

content\_copy

1. En Cloud Shell, ingresa el siguiente comando para clonar el repositorio en el entorno de Cloud Shell del lab.

git clone https://github.com/GoogleCloudPlatform/training-data-analyst

Se copió correctamente

content\_copy

1. Crea un vínculo simbólico como un acceso directo al directorio de trabajo:

ln -s ~/training-data-analyst/courses/ak8s/v1.1 ~/ak8s

Se copió correctamente

content\_copy

1. Cambia al directorio que contenga los archivos de muestra de este lab:

cd ~/ak8s/Deployments/

Se copió correctamente

content\_copy

**Crea un manifiesto de implementación**

Crearás una implementación con el manifiesto de muestra denominado nginx-deployment.yaml que se te proporcionó. Esta implementación está configurada para ejecutar tres réplicas de Pod con un solo contenedor nginx en cada Pod que escucha en el puerto TCP 80:

apiVersion: apps/v1

kind: Deployment

metadata:

name: nginx-deployment

labels:

app: nginx

spec:

replicas: 3

selector:

matchLabels:

app: nginx

template:

metadata:

labels:

app: nginx

spec:

containers:

- name: nginx

image: nginx:1.7.9

ports:

- containerPort: 80

Se copió correctamente

content\_copy

1. Para implementar tu manifiesto, ejecuta el siguiente comando:

kubectl apply -f ./nginx-deployment.yaml

Se copió correctamente

content\_copy

1. Para ver una lista de implementaciones, ejecuta el siguiente comando:

kubectl get deployments

Se copió correctamente

content\_copy

El resultado debería verse como este ejemplo.

**Resultado:**

NAME READY UP-TO-DATE AVAILABLE AGE

nginx-deployment 0/3 3 0 3s

1. Espera unos segundos y repite el comando hasta que la cantidad de implementaciones en CURRENT informada por el comando coincida con la cantidad de implementaciones indicada en DESIRED.

El resultado final debe verse como el ejemplo.

**Resultado:**

NAME READY UP-TO-DATE AVAILABLE AGE

nginx-deployment 3/3 3 3 42s

Haz clic en *Revisar mi progreso* para verificar el objetivo.

Crear e implementar el manifiesto nginx-deployment

Revisar mi progreso

**Tarea 2. Aumenta y reduce manualmente la escala de la cantidad de Pods en las implementaciones**

A veces, deseas cerrar una instancia de Pod. Otras veces, quieres contar con diez Pods en ejecución. En Kubernetes, puedes escalar un Pod específico para alcanzar la cantidad de instancias deseada. Para cerrarlos, implementa una reducción de escala a cero.

En esta tarea, aumentarás y reducirás la escala de los Pods en la consola de Google Cloud y Cloud Shell.

**Aumenta y reduce la escala de los Pods en la consola**

1. Cambia a la pestaña de la consola de Google Cloud.
2. En **Menú de navegación** (Ícono del menú de navegación), haz clic en **Kubernetes Engine** > **Cargas de trabajo**.
3. Haz clic en **nginx-deployment** (tu implementación) para abrir la página Detalles de la implementación.
4. En la parte superior, haz clic en **ACCIONES > Escalar > Editar Réplicas**.
5. Escribe **1** y haz clic en **ESCALAR**.

Esta acción reduce tu clúster. Deberías ver el estado del Pod que se actualiza en **Pods administrados**. Es posible que tengas que hacer clic en **Actualizar**.

**Aumenta y reduce la escala de los Pods en la shell**

1. Regresa a la pestaña del navegador de Cloud Shell.
2. En Cloud Shell, ejecuta el siguiente comando para ver una lista de Pods en las implementaciones:

kubectl get deployments

Se copió correctamente

content\_copy

**Resultado:**

NAME READY UP-TO-DATE AVAILABLE AGE

nginx-deployment 1/1 1 1 3m

1. Para escalar el Pod hasta tres réplicas, ejecuta el siguiente comando:

kubectl scale --replicas=3 deployment nginx-deployment

Se copió correctamente

content\_copy

1. Para ver una lista de Pods en las implementaciones, ejecuta el siguiente comando:

kubectl get deployments

Se copió correctamente

content\_copy

**Resultado:**

NAME READY UP-TO-DATE AVAILABLE AGE

nginx-deployment 3/3 3 3 4m

**Tarea 3. Activa un lanzamiento y una reversión de la implementación**

El lanzamiento de una implementación se activa solo si se modifica la plantilla del Pod de la implementación (es decir, .spec.template), por ejemplo, si se actualizan las etiquetas o las imágenes de contenedor de la plantilla. Otras actualizaciones, como escalar la implementación, no activan un lanzamiento.

En esta tarea, activarás el lanzamiento de la implementación y, a continuación, activarás tu reversión.

**Activa el lanzamiento de una implementación**

1. Para actualizar la versión de nginx en la implementación, ejecuta el siguiente comando:

kubectl set image deployment.v1.apps/nginx-deployment nginx=nginx:1.9.1 --record

Se copió correctamente

content\_copy

Esto actualiza la imagen de contenedor en tu implementación a nginx v1.9.1.

1. Para ver el estado del lanzamiento, ejecuta el siguiente comando:

kubectl rollout status deployment.v1.apps/nginx-deployment

Se copió correctamente

content\_copy

El resultado debería verse como el ejemplo.

**Resultado:**

Waiting for rollout to finish: 1 out of 3 new replicas updated…

Waiting for rollout to finish: 1 out of 3 new replicas updated…

Waiting for rollout to finish: 1 out of 3 new replicas updated…

Waiting for rollout to finish: 2 out of 3 new replicas updated…

Waiting for rollout to finish: 2 out of 3 new replicas updated…

Waiting for rollout to finish: 2 out of 3 new replicas updated…

Waiting for rollout to finish: 1 old replicas pending termination…

Waiting for rollout to finish: 1 old replicas pending termination…

deployment "nginx-deployment" successfully rolled out

1. Para verificar el cambio, obtén la lista de implementaciones:

kubectl get deployments

Se copió correctamente

content\_copy

El resultado debería verse como el ejemplo.

**Resultado:**

NAME READY UP-TO-DATE AVAILABLE AGE

nginx-deployment 3/3 3 3 6m

Haz clic en *Revisar mi progreso* para verificar el objetivo.

Actualizar la versión de nginx en la implementación

Revisar mi progreso

1. Consulta el historial de lanzamientos de la implementación.

kubectl rollout history deployment nginx-deployment

Se copió correctamente

content\_copy

El resultado debería verse como el ejemplo. Es posible que tu resultado no sea exactamente igual.

**Resultado:**

deployments "nginx-deployment"

REVISION CHANGE-CAUSE

1

2 kubectl set image deployment.v1.apps/nginx-deployment nginx=nginx:1.9.1 --record=true

Activa una reversión de la implementación

Para revertir el lanzamiento de un objeto, puedes usar el comando kubectl rollout undo.

1. Para volver a la versión anterior de la implementación de nginx, ejecuta el siguiente comando:

kubectl rollout undo deployments nginx-deployment

Se copió correctamente

content\_copy

1. Ve el historial de lanzamientos actualizado de la implementación:

kubectl rollout history deployment nginx-deployment

Se copió correctamente

content\_copy

El resultado debería verse como el ejemplo. Es posible que tu resultado no sea exactamente igual.

**Resultado:**

deployments "nginx-deployment"

REVISION CHANGE-CAUSE

2 kubectl set image deployment.v1.apps/nginx-deployment nginx=nginx:1.9.1 --record=true

3

1. Ve los detalles de la última revisión de la implementación:

kubectl rollout history deployment/nginx-deployment --revision=3

Se copió correctamente

content\_copy

El resultado debería verse como el ejemplo. Es posible que tu resultado no sea exactamente igual, pero mostrará que la revisión actual se revirtió a nginx:1.7.9.

**Resultado:**

deployments "nginx-deployment" with revision #3

Pod Template:

Labels: app=nginx

pod-template-hash=3123191453

Containers:

nginx:

Image: nginx:1.7.9

Port: 80/TCP

Host Port: 0/TCP

Environment:

Mounts:

Volumes:

**Tarea 4. Define el tipo de servicio en el manifiesto**

En esta tarea, crearás y verificarás un servicio que controla el tráfico entrante a una aplicación. Los servicios se pueden configurar como ClusterIP, NodePort o LoadBalancer. En este lab, configurarás un LoadBalancer.

**Define los tipos de servicio en el manifiesto**

Se te proporcionó un archivo de manifiesto llamado service-nginx.yaml que implementa un tipo de servicio LoadBalancer. Este servicio está configurado para distribuir el tráfico entrante en el puerto TCP 60000 al puerto 80 en cualquier contenedor que tenga la etiqueta app: nginx.

apiVersion: v1

kind: Service

metadata:

name: nginx

spec:

type: LoadBalancer

selector:

app: nginx

ports:

- protocol: TCP

port: 60000

targetPort: 80

Se copió correctamente

content\_copy

* En Cloud Shell, ejecuta el siguiente comando para implementar tu manifiesto:

kubectl apply -f ./service-nginx.yaml

Se copió correctamente

content\_copy

Este manifiesto define un servicio y lo aplica a Pods que corresponden al selector. En este caso, el manifiesto se aplica al contenedor nginx que implementaste en la tarea 1. Este servicio también se aplica a cualquier otro pod con la etiqueta app: nginx, incluidos los que se crean después del servicio.

**Verifica la creación de LoadBalancer**

1. Para ver los detalles del servicio de nginx, ejecuta el siguiente comando:

kubectl get service nginx

Se copió correctamente

content\_copy

El resultado debería verse como el ejemplo.

**Resultado:**

NAME CLUSTER\_IP EXTERNAL\_IP PORT(S) SELECTOR AGE

nginx 10.X.X.X X.X.X.X 60000/TCP run=nginx 1m

1. Cuando aparezca la IP externa, abre http://[IP\_EXTERNA]:60000/ en una pestaña del navegador nueva para ver cómo el servidor recibe contenido a través del balanceo de cargas de red.

**Nota:**Es posible que los datos del campo **ExternalIP** de tu servicio demoren algunos segundos en propagarse. Esto es normal. Simplemente vuelve a ejecutar el comando kubectl get services nginx cada algunos segundos hasta que los datos del campo se propaguen.

Haz clic en *Revisar mi progreso* para verificar el objetivo.

Implementar el archivo de manifiesto que implementa el tipo de servicio LoadBalancer

Revisar mi progreso

**Tarea 5. Realiza una implementación de versiones canary**

Una implementación de versiones canary es una implementación separada que se utiliza para probar una nueva versión de tu aplicación. Un solo servicio apunta a las implementaciones de versiones canary y a las normales. Además, puede dirigir un subconjunto de usuarios a la versión canary para mitigar el riesgo de nuevos lanzamientos.

El archivo de manifiesto nginx-canary.yaml que se te proporciona implementa un solo Pod que ejecuta una versión más reciente de nginx que tu implementación principal. En esta tarea, crearás una implementación de versiones canary con este archivo de implementación nuevo:

apiVersion: apps/v1

kind: Deployment

metadata:

name: nginx-canary

labels:

app: nginx

spec:

replicas: 1

selector:

matchLabels:

app: nginx

template:

metadata:

labels:

app: nginx

track: canary

Version: 1.9.1

spec:

containers:

- name: nginx

image: nginx:1.9.1

ports:

- containerPort: 80

Se copió correctamente

content\_copy

El manifiesto para el servicio de nginx que implementaste en la tarea anterior utiliza un selector de etiquetas a fin de apuntar a los Pods con la etiqueta app: nginx. Tanto la implementación normal como esta nueva implementación de versiones canary tienen la etiqueta app: nginx. El servicio distribuirá las conexiones entrantes a los Pods de implementación normal y de versiones canary. La implementación de versiones canary tiene menos réplicas (Pods) que la implementación normal y, por lo tanto, está disponible para menos usuarios que esta última.

1. Crea la implementación de versiones canary basada en el archivo de configuración:

kubectl apply -f nginx-canary.yaml

Se copió correctamente

content\_copy

1. Cuando se complete la implementación, verifica que estén presentes las implementaciones nginx y nginx-canary:

kubectl get deployments

Se copió correctamente

content\_copy

1. Regresa a la pestaña del navegador que está conectada a la IP externa del servicio LoadBalancer y actualiza la página. Deberías seguir viendo la página estándar Welcome to nginx.
2. Regresa a Cloud Shell y reduce verticalmente la escala de la implementación principal a 0 réplicas:

kubectl scale --replicas=0 deployment nginx-deployment

Se copió correctamente

content\_copy

1. Verifica que la única réplica en ejecución sea ahora la implementación de versiones canary:

kubectl get deployments

Se copió correctamente

content\_copy

1. Regresa a la pestaña del navegador que está conectada a la IP externa del servicio LoadBalancer y actualiza la página. Deberías seguir viendo la página estándar Welcome to nginx, que muestra que el servicio está balanceando automáticamente el tráfico a la implementación de versiones canary.

Haz clic en *Revisar mi progreso* para verificar el objetivo.

Crear una implementación de versiones canary

Revisar mi progreso

**Afinidad de sesión**

La configuración del servicio utilizada en el lab no garantiza que todas las solicitudes de un solo cliente siempre se conectarán al mismo Pod. Cada solicitud se trata por separado y se puede conectar a la implementación nginx normal o a la implementación nginx-canary.

Esta posibilidad de hacer cambios entre diferentes versiones puede causar problemas si hay modificaciones significativas en la funcionalidad del lanzamiento Canary. Para evitar esto, puedes establecer el campo sessionAffinity como ClientIP en la especificación del servicio si necesitas la primera solicitud de un cliente para determinar qué Pod se utilizará para todas las conexiones posteriores.

Por ejemplo:

apiVersion: v1

kind: Service

metadata:

name: nginx

spec:

type: LoadBalancer

sessionAffinity: ClientIP

selector:

app: nginx

ports:

- protocol: TCP

port: 60000

targetPort: 80

Se copió correctamente

content\_copy

**Finalice su lab**

Cuando haya completado el lab, haga clic en **Finalizar lab**. Google Cloud Skills Boost quitará los recursos que usó y limpiará la cuenta.

Tendrá la oportunidad de calificar su experiencia en el lab. Seleccione la cantidad de estrellas que corresponda, ingrese un comentario y haga clic en **Enviar**.

La cantidad de estrellas indica lo siguiente:

* 1 estrella = Muy insatisfecho
* 2 estrellas = Insatisfecho
* 3 estrellas = Neutral
* 4 estrellas = Satisfecho
* 5 estrellas = Muy satisfecho

Puede cerrar el cuadro de diálogo si no desea proporcionar comentarios.

Para enviar comentarios, sugerencias o correcciones, use la pestaña **Asistencia**.

Copyright 2020 Google LLC. All rights reserved. Google y el logotipo de Google son marcas de Google LLC. Los demás nombres de produc