

METODO 1. //Nota del docente. este método 1 es el método largo copiando y pegando cada instrucción. Abajo al final de este método hay un segundo método pero debe ver el video antes de usar el método 2. El video se encuentra en la carpeta compartida y se llama: Videob 14 Clase virtual GOOGLE CLOUD COMPUTING FOUNDATIONS miércoles 04 dic 2024.

Estos pasos siempre hay que hacerlos en todos los labs de primero. copiar y pegar.

```
gcloud auth list  
gcloud config list project  
gcloud config set compute/region Region  
gcloud config set compute/zone Zone  
=====
```

PASO 1 ### CREAR Y ASINGAR A VARIABLES GLOBALES //nota del profesor: aquí sustituir por los valores del laboratorio. los valores que aparecen aquí fueron los que yo use para mi último lab debe usar los valores que a usted le apare en el panel izquierdo donde esta el Id de proyecto y al contraseña. El numero de puerto no se muestra: poner 80. Estas variables gloabales no evitan tener que estar haciendo sustituciones en el bloc de notas se hacen automáticas abajo en los códigos donde se muestran con signo de dolar usted no tiene que hacer las sustituciones. simplemente pega el código de abajo tal como está.

```
export INSTANCE=nucleus-jumphost-326  
export PORT_NO=80  
export FIREWALL=allow-tcp-rule-549  
export REGION=us-central1  
export ZONE=us-central1-c  
=====
```

PASO 2. CREAR UNA INSTANCIA CON CÓDIGO. //También se podría crear desde la consola en Compute Engie-Virtual Machines-Crear Instance cuidado si lo hace desde la consola debe usar el tipo de máquina e2-micro. Recuerde que en Linux, la barra inversa o back slash "\" se usa para continuar comandos que son demasiado largos para caber en una sola línea, esto nos permite poder partir un comando largo en pequeñas lineas como se muestra abajo. cuidado no puede omitirlas. Si desea omitirlas tendra que pegar todos los trozos y hacer una solo comando largo. Por ejemplo el siguiente comando sin las barras quedaría así. gcloud compute instances create \$INSTANCE --zone=\$ZONE --machine-type=e2-micro. Pero simplemente copie y pegue el los siguientes códidos tales como están. //No hace falta sustituir la INSTANCE ni la ZONE porque la variable es global y ya lo hicimos arriba.

```
gcloud compute instances create $INSTANCE \  
--zone=$ZONE \  
--machine-type=e2-micro  
=====
```

PASO 3. CRER UN CLUSTER SERVICE KUBERNETES. Esto lleva 5 subpasos. //Aquí no hace falta sustituir nada.

3.1 Crear un servidor web NGINX.

```
cat << EOF > startup.sh  
#!/bin/bash  
apt-get update  
apt-get install -y nginx  
service nginx start  
sed -i -- 's/nginx/Google Cloud Platform - '\''"\$HOSTNAME"'\' /'  
/var/www/html/index.nginx-debian.html
```

EOF

3.2 Crear un cluster de GKE. Un clúster consta de, al menos, una máquina de instancia principal del clúster y varias máquinas trabajadoras, llamadas nodos. Los nodos son instancias de máquinas virtuales (VM) de Compute Engine que ejecutan los procesos de Kubernetes necesarios para que sean parte del clúster. //No hace falta sustituir la ZONE porque la variable es global y ya lo hicimos arriba.

```
gcloud container clusters create nucleus-backend \
    --num-nodes=1 \
    --zone=$ZONE
=====
```

3.3 Solicitar las credenciales. Después de crear tu clúster, necesitas credenciales de autenticación para interactuar con él. //No hace falta sustituir la ZONE porque la variable es global y ya lo hicimos arriba.

```
gcloud container clusters get-credentials nucleus-backend \
    --zone $ZONE
=====
```

3.4 Implementa una aplicación en el clúster. Ahora puedes implementar una aplicación alojada en contenedores en el clúster. En este lab, ejecutarás hello-app en tu clúster.

```
kubectl create deployment hello-server \
    --image=gcr.io/google-samples/hello-app:2.0
=====
```

3.5 Crear un objeto Service. Esto es un recurso de Kubernetes que te permite exponer tu aplicación al tráfico externo. //No hace falta sustituir la PORT_NO porque la variable es global y ya lo hicimos arriba.

```
kubectl expose deployment hello-server \
    --type=LoadBalancer \
    --port $PORT_NO
=====
=====
```

PASO 4. CONFIGURAR UN BALANCEADOR DE CARGAS HTTP. Esto lleva 5 subpasos.

4.1 Crear una plantilla de instancias. //No hace falta sustituir la REGION porque la variable es global y ya lo hicimos arriba.

```
gcloud compute instance-templates create web-server-template \
    --metadata-from-file startup-script=startup.sh \
    --machine-type g1-small \
    --region $REGION
=====
=====
```

4.2 Crear un grupo de instancias administrado basado en la plantilla //No hace falta sustituir la REGION porque la variable es global y ya lo hicimos arriba. Cada linea de guiones es una instrucción independiente pero algunos pasos tienen varias instrucciones.

```
gcloud compute target-pools create nginx-pool --region $REGION
=====
gcloud compute instance-groups managed create web-server-group \
=====
```

```
--base-instance-name web-server \
--size 2 \
--template web-server-template \
--region $REGION
```

4.3 Crear una regla de firewall llamada Firewall rule para permitir el tráfico (80/tcp) //No hace falta sustituir la FIREWALL porque la variable es global y ya lo hicimos arriba.

```
gcloud compute firewall-rules create $FIREWALL \
    --allow tcp:80
```

4.4 Crear una verificación de estado de salud/health

```
gcloud compute http-health-checks create http-basic-check
```

4.5 Crear un servicio de backend y agrega tu grupo de instancias como backend al grupo de servicios de backend con el puerto nombrado (http:80). //No hace falta sustituir la REGION porque la variable es global y ya lo hicimos arriba.

```
gcloud compute instance-groups managed \
    set-named-ports web-server-group \
    --named-ports http:80 \
    --region $REGION
```

```
gcloud compute backend-services create web-server-backend \
    --protocol HTTP \
    --http-health-checks http-basic-check \
    --global
```

```
gcloud compute backend-services add-backend web-server-backend \
    --instance-group web-server-group \
    --instance-group-region $REGION \
    --global
```

4.6 Crea un mapa de URL y orienta el proxy HTTP para enrutar las solicitudes entrantes al servicio de backend predeterminado.

```
gcloud compute url-maps create web-server-map \
    --default-service web-server-backend
```

```
gcloud compute target-http-proxies create http-lb-proxy \
    --url-map web-server-map
```

4.7 Crea un Proxy HTTP de destino para enrutar las solicitudes a tu mapa de URLs. //No hace falta sustituir la FIREWALL porque la variable es global y ya lo hicimos arriba.

```
gcloud compute forwarding-rules create $FIREWALL \
    --global \
    --target-http-proxy http-lb-proxy \
    --ports 80
```

4.8 Crea una regla de reenvío

```
gcloud compute forwarding-rules list
```

Es posible que debas esperar entre 5 y 7 minutos para recibir la puntuación de esta tarea.

FIN DEL METODO UNO.

```
#####
#####
```

METODO DOS CONFIGURA: Configura las variables Globales en el shell y luego pega el código de abajo todo se hace con un solo paso simplemente espera y ve haciendo clic en verificaciones del lab respectivo. Este paso lo mostré en el video de la noche del miércoles 04 de diciembre en la carpeta compartida se muestra como: Videob 14 Clase virtual GOOGLE CLOUD COMPUTING FOUNDATIONS miércoles 04 dic 2024.
Recuerda que al final debe esperar más o menos 5 o 7 minutos para poder verificar el último paso y le aparezcan los 100 puntos. Ver del video.

Por favor ver el video antes de usar este segundo método

```
### Assign Variables
```

```
-----  
export INSTANCE=nucleus-jumphost-326  
export PORT_NO=80  
export FIREWALL=allow-tcp-rule-549  
export REGION=us-central1  
export ZONE=us-central1-c  
-----
```

```
curl -LO raw.githubusercontent.com/QUICK-GCP-LAB/2-Minutes-Labs-Solutions/main/  
Create%20and%20Manage%20Cloud%20Resources%20Challenge%20Lab/gsp313.sh
```

```
sudo chmod +x gsp313.sh
```

```
./gsp313.sh
```

Es posible que debas esperar entre 5 y 7 minutos para recibir la puntuación de esta tarea.