

CLASE 49

Elaborado por: M. en C. Ukranio Coronilla

Para tener como aprobada la presente práctica será necesario adjuntar todos los ítems numerados que se le soliciten en un único archivo pdf el cual será enviado de manera individual a la plataforma teams.

Como se explica en el texto del curso en el subtema 10.9, existen arquitecturas de aprovisionamiento rápido que automatizan el aprovisionamiento de recursos de TI mediante el uso de scripts y templates (plantillas).

En el curso hemos experimentado que estar creando instancias, subir el código, y ejecutarlo puede ser propenso a errores además de consumir tiempo, sobre todo si se trata de crear decenas de instancias.

En esta práctica vamos a automatizar la creación de varias instancias mediante el uso de una plantilla de instancia, y haremos uso de las tecnologías de escalamiento dinámico y la reparación automática de instancias que nos brinda la nube de Google.

Plantillas de instancia

Acceda a su consola GCP dando click en el link:

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

Para crear nuestra plantilla de instancias le damos click a la hamburguesa ☰ en la esquina superior izquierda para ver el panel (dashboard) y seleccionamos: Compute Engine-> Plantillas de Instancia.

Posteriormente damos click en [+ CREAR PLANTILLA DE INSTANCIAS](#) y en el apartado nombre le daremos un nombre valido para nuestra plantilla, seleccionamos el tipo de máquina que queramos, por ejemplo, de la serie E2 la e2-micro, permitimos el tráfico HTTP y le damos click a la flecha que abre el menú de Opciones avanzadas:

Firewall ⓘ

Agrega etiquetas y reglas de firewall para permitir determinados tipos de tráfico de red desde Internet

- ☒ Permitir tráfico HTTP
- ☐ Permitir tráfico HTTPS
- ☐ Permitir las verificaciones de estado del balanceador de cargas

Opciones avanzadas

Networking, disks, security, management, sole-tenancy



Del menú que aparece le damos click en la flecha que abre el menú de Administración, con lo que aparece:

Administración

Descripción, protección contra la eliminación, reservas y automatización

Descripción

Reservas

Usar de forma automática la reserva creada

Cuando crees una instancia a partir de esta plantilla, usa una reserva existente

Automatización

Secuencia de comandos de inicio

Puedes especificar una secuencia de comandos de inicio que se ejecutará cuando tu instancia se inicie o se reinicie. Las secuencias de comandos de inicio se pueden usar para instalar software y actualizaciones, y también para garantizar que los servicios se ejecuten dentro de la máquina virtual. [Más información](#)

Dentro del campo **Automatización** vamos a colocar un script que se va a ejecutar automáticamente cada que se inicie o reinicie nuestra instancia. Se sugiere el siguiente script que se explica a continuación (cambie el nombre del bucket en amarillo por el suyo):


```
#!/bin/bash
apt-get update
apt-get -y -force=yes install openjdk-17-jre-headless
gcloud storage cp gs://application-bucket-ukranio/*.jar ./
java -jar my-app-1.0-SNAPSHOT-jar-with-dependencies.jar 80
```

La primera línea indica que estaremos utilizando el intérprete de comandos bash. La línea 2 y 3 es la que hemos utilizado para instalar el JRE de openjdk salvo que le agregamos un par de opciones para que no pregunte si se desea realizar la instalación y lo automatice. La línea 4 llama al programa gcloud para copiar el archivo jar del bucket a la carpeta home de la instancia creada, y la línea 5 ejecuta el archivo jar en el puerto 80.

Por último, damos click en **CREAR**, y entonces nos aparece la plantilla creada en la lista de plantillas disponibles. Saque una captura de la plantilla creada (**Item 1**)

Vamos a probar nuestra plantilla de instancias creando una instancia para verificar que funciona correctamente. Al final del renglón de la plantilla creada damos click en los tres puntos y seleccionamos **Crear VM**:

Filtro Filtrar plantillas de instancias

<input type="checkbox"/>	Nombre ↑	Tipo de máquina	Imagen	Tipo de disco	Ubicación ?	Política de posición ?	En uso por	Acciones
<input type="checkbox"/>	instance-template-ukranio	e2-micro	debian-12-bookworm-v20240617	Disco persistente equilibrado	us-central1	No hay políticas		

No modificamos nada más sólo le damos click en **CREAR** en la parte inferior de la página, con lo cual se crea la instancia y se ejecuta el script, lo cual tarda unos dos minutos en el setup de la instancia, instalar el runtime de java y ejecutar el archivo jar por lo que tendrá que esperar a que eso suceda. Pasado el tiempo podemos copiar la dirección IP de la instancia creada, y usarla en un navegador para acceder al servicio web, pruébelo.

Es posible que el sitio no se ejecute correctamente dentro de la red Wifi de la ESCOM, pero en su celular usando los datos debería funcionar sin problemas.

Al terminar de hacer la prueba es importante borrar la instancia que acabamos de crear y no la plantilla de instancias la cual no tiene ningún costo asociado.

Grupos de instancias

Ahora vamos a crear un grupo de instancias a partir de la plantilla de instancias, con lo cual podremos administrar un cluster de instancias, monitorearlas, añadir más instancias durante los picos de tráfico y arreglar alguna falla en el cluster.

Para ello vamos a la plantilla de instancias que creamos, damos click en los tres puntos, pero ahora damos click en **Crear grupo de instancias**.

Filtro Filtrar plantillas de instancias

<input type="checkbox"/>	Nombre ↑	Tipo de máquina	Imagen	Tipo de disco	Ubicación ?	Política de posición ?	En uso por	Acciones
<input type="checkbox"/>	instance-template-ukranio	e2-micro	debian-12-bookworm-v20240617	Disco persistente equilibrado	us-central1	No hay políticas		

 Crear VM
 Crear grupo de instancias
 Borrar

Le damos un nombre al grupo, y en **Ubicación** seleccionamos **Varias zonas** para distribuir nuestras instancias en múltiples zonas dentro de la región, con lo cual

garantizamos alta disponibilidad aún si una de las zonas no está disponible debido a algún problema. Seleccionamos una región y damos click a todas las posibles zonas:

Ubicación

Para aumentar la disponibilidad, selecciona varias zonas de una región en lugar de una sola. [Más información](#)

- ☐ Zona única
- ☒ Varias zonas

Región *
us-central1 (Iowa)

Forma de distribución objetivo
Uniforme

Instance redistribution ?
☒ Allow instance redistribution

Zonas ?

- ☒ us-central1-c
- ☒ us-central1-a
- ☒ us-central1-f
- ☒ us-central1-b

CANCELAR ACEPTAR

En **Ajuste de escala automático** verificamos que esté en **Activado: agrega y quita instancias del grupo**, el **Numero mínimo de instancias** se recomienda sea al menos el mismo número de zonas que en mi caso son cuatro, pero sólo vamos a dejar dos y en máximo vamos a poner seis. En **Autoscaling signals** dejamos la configuración predeterminada de 60% con lo cual al usarse más de 60% de CPU provocará un escalamiento hacia arriba o al decrecer por debajo de 60% un escalamiento hacia abajo, quedando como sigue:

Ajuste de escala automático

Usa el ajuste de escala automático para agregar y quitar instancias de forma automática en el grupo durante los períodos de cargas altas y bajas. [Más información](#)

Modo de ajuste de escala automático
Activado: agrega y quita instancias del grupo

Número mínimo de instancias *
2 ?

Número máximo de instancias *
6 ?

! Para maximizar la disponibilidad, la cantidad mínima de instancias debe ser al menos igual a la cantidad de zonas. Se colocarán instancias adicionales en diferentes zonas. [Cómo distribuir instancias mediante grupos de instancias administrados regionales](#)

Autoscaling signals

Usa los indicadores para determinar cuándo escalar el grupo. [Más información](#)

✓ Uso de CPU: 60% (configuración predeterminada)
El ajuste de escala automático predictivo está off

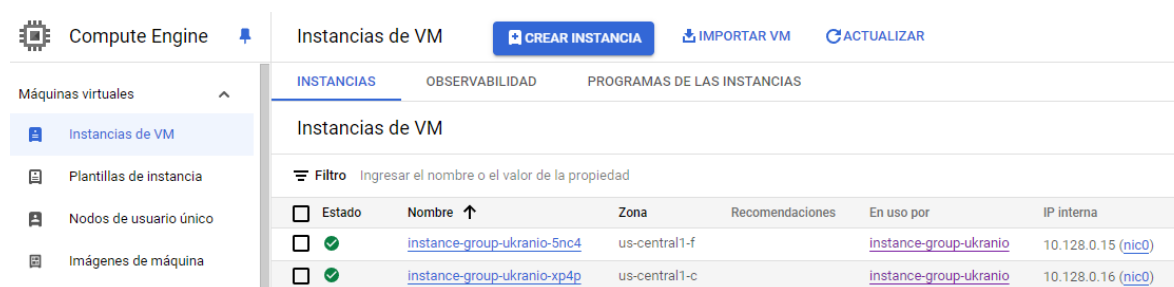
En **Periodo de inicialización** va el número de segundos que el escalador automático debería esperar después de que una máquina virtual ha iniciado, antes de que el escalador comience a obtener información nueva del uso de CPU. En nuestro caso

ponemos cinco minutos (300 segundos) para evitar hacer un monitoreo en el momento que se levanten automáticamente una o varias instancias, pues podría estarse consumiendo mucho CPU y se decida levantar más instancias de manera recursiva.

Al final damos click en **CREAR** en la parte inferior izquierda de la ventana, con lo que se crea nuestro grupo de instancias comenzando con dos instancias.

A este momento ya habrá podido observar que cada que se le pide a la plataforma reservar o eliminar algún recurso, aparece una notificación en el icono 🔔 de la parte superior derecha, el cual al darle click nos muestra la información relacionada.

Si observamos en Compute Engine->Instancias de VM podemos ver las dos instancias creadas que forman parte del grupo, las cuales se ejecutan en distintas zonas:



Compute Engine		Instancias de VM	CREAR INSTANCIA	IMPORTAR VM	ACTUALIZAR																		
Máquinas virtuales		INSTANCIAS OBSERVABILIDAD PROGRAMAS DE LAS INSTANCIAS																					
Instancias de VM		Instancias de VM																					
Plantillas de instancia		Filtro Ingresar el nombre o el valor de la propiedad																					
Nodos de usuario único		<table><thead><tr><th>Estado</th><th>Nombre ↑</th><th>Zona</th><th>Recomendaciones</th><th>En uso por</th><th>IP interna</th></tr></thead><tbody><tr><td><input type="checkbox"/></td><td>instance-group-ukranio-5nc4</td><td>us-central1-f</td><td></td><td>instance-group-ukranio</td><td>10.128.0.15 (nic0)</td></tr><tr><td><input type="checkbox"/></td><td>instance-group-ukranio-xp4p</td><td>us-central1-c</td><td></td><td>instance-group-ukranio</td><td>10.128.0.16 (nic0)</td></tr></tbody></table>				Estado	Nombre ↑	Zona	Recomendaciones	En uso por	IP interna	<input type="checkbox"/>	instance-group-ukranio-5nc4	us-central1-f		instance-group-ukranio	10.128.0.15 (nic0)	<input type="checkbox"/>	instance-group-ukranio-xp4p	us-central1-c		instance-group-ukranio	10.128.0.16 (nic0)
Estado	Nombre ↑	Zona	Recomendaciones	En uso por	IP interna																		
<input type="checkbox"/>	instance-group-ukranio-5nc4	us-central1-f		instance-group-ukranio	10.128.0.15 (nic0)																		
<input type="checkbox"/>	instance-group-ukranio-xp4p	us-central1-c		instance-group-ukranio	10.128.0.16 (nic0)																		
Imágenes de máquina																							

Si esperamos dos minutos, podemos probar que las IPs de ambas instancias ya ejecutan nuestra aplicación web(pruébelos), y después de cinco minutos ya se estarán monitoreando las instancias para verificar su uso de CPU.

Saque una captura de las dos instancias que forman parte del grupo (**Item 2**).

Para administrar el grupo de instancias damos click en el nombre del grupo de una de las dos instancias creadas:

Instancias de VM

Filtro Ingresar el nombre o el valor de la propiedad

Estado	Nombre ↑	Zona	Recomendaciones	En uso por	IP interna
<input type="checkbox"/>	instance-group-ukranio-m55z	us-central1-a		instance-group-ukranio	10.128.0.33 (nic0)
<input type="checkbox"/>	instance-group-ukranio-njl5	us-central1-f		instance-group-ukranio	10.128.0.34 (nic0)

En la nueva ventana podemos observar diversas pestañas para administrar el grupo. En la parte superior la descripción general, detalles, supervisión y errores, y en la parte baja para suspender, detener, iniciar/reanudar, quitar del grupo o borrar el grupo de instancias:

DESCRIPCIÓN GENERAL DETALLES SUPERVISIÓN ERRORES

Instancias por condición
2 instancias ⓘ
✓ 2

Instancia por estado ⓘ
No configurada
Reparación automática desactivada. [Configurar](#)

Ajuste de escala automático
Activo (min 2, max 6)
Based on 1 métrica and 0 programas ⓘ

Status ✓ Ready
Creation Time jun 19, 2024, 8:48:07 a. m. UTC-06:00
Description
Target running size 2
Template [instance-template-ukranio \(Regional\)](#)
Location us-central1 (4/4)

Instancias de VM

⏸ SUSPENDER ■ DETENER ▶ INICIAR/REANUDAR ⌵ QUITAR DEL GRUPO 🗑 BORRAR

Filtro Ingresar el nombre o el valor de la propiedad ⓘ

Estado	Nombre ↑	Fecha/hora de creación	Plantilla	Zona	Configuración por instancia	IP interna	Conectar
--------	----------	------------------------	-----------	------	-----------------------------	------------	----------

Mas abajo también podrá ver las dos instancias activas.

Haga una captura de pantalla del gráfico **tiempo vs instancias** dando click a [SUPERVISIÓN](#).

Escalamiento

Para ver el escalamiento en acción generaremos carga en una de las instancias para que consuma más del 60% de CPU. Esto lo lograremos con una aplicación que hace uso intensivo de CPU conocida como stress. Abrimos una terminal ssh en una de las instancias, e instalamos la aplicación stress con:

```
sudo apt-get install stress
```

Corremos el comando **top**, para verificar que prácticamente no hay uso de CPU

corremos el comando stress con:

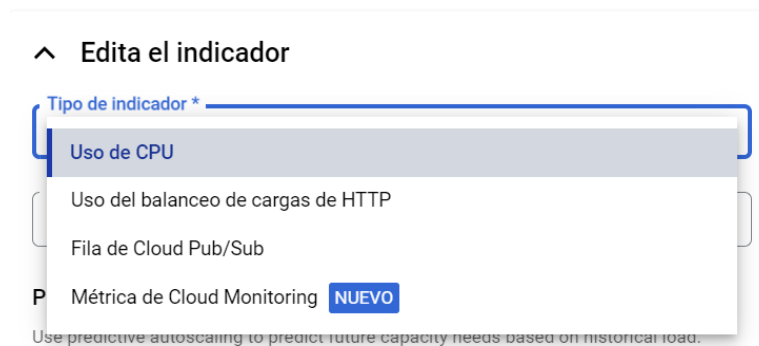
```
stress --cpu 2 --timeout 60s &
```

Verificamos con el comando top que efectivamente se está haciendo uso intensivo de CPU y posteriormente observamos en Compute Engine->Instancias de VM que ya se han creado automáticamente más instancias con el servidor web funcionando. Si esperamos unos minutos veremos que al dejar de usarse el CPU nuevamente vuelven a eliminarse las instancias creadas para quedar de nuevo en las dos iniciales.

Volvemos al administrador del grupo de instancias y vemos como están los gráficos **tiempo vs instancias** , **Utilización del escalador automático (CPU)** y **Uso de CPU** dando click a [SUPERVISIÓN](#) y posteriormente mandamos una captura de pantalla de estos gráficos (**Item 3**).

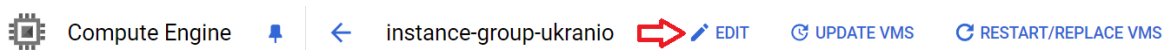
En el comando stress que se acaba de ejecutar el 2 indica que correrá dos instancias de la aplicación, el tiempo en segundos que se estará ejecutando lo indica en la opción timeout. El ampersand pide ejecutar el comando en segundo plano, por lo que podemos ejecutar nuevamente el comando top para visualizar que se usa casi el 100% de CPU.

Lo que acabamos de ver es un ejemplo de escalamiento dinámico basado en uso de CPU, pero si desea puede cambiar el indicador del grupo de instancias al dar click en [EDIT](#) y entonces podrá seleccionar otro tipo de indicador:

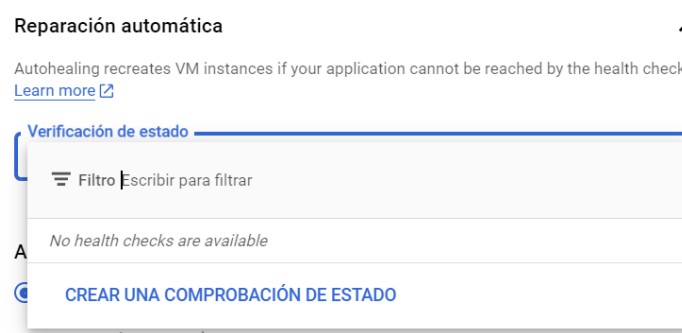


Autohealing

Ahora veremos la reparación automática (autohealing). Vamos al administrador del grupo de instancias y damos click en EDIT:



Nos deslizamos en la página hasta llegar a **Reparación automática**, damos click en Verificación de estado y seleccionamos **CREAR UNA COMPROBACIÓN DE ESTADO**:



Aparece una ventana de **Verificación de estado** donde vamos a indicarle la manera de saber si una instancia está dañada, le damos un nombre, indicamos el protocolo como HTTP, el puerto 80 y el endpoint status:

Verificación de estado

Nombre *

aplicacion-web-healthcheck

?

Minúsculas, sin espacios.

Descripción

Alcance

☒ Global

☐ Regional

Protocolo

HTTP

Puerto *

80

?

Protocolo de proxy

NINGUNO

Ruta de la solicitud *

/status

?

Abajo en los **Criterios de buen estado** ponemos que verifique cada 10 segundos, que espere 5 segundos por una respuesta del endpoint /status, que haga 3 chequeos consecutivos a partir de los cuales se considera saludable la instancia y 3 chequeos consecutivos a partir de los cuales se considera no saludable la instancia:

Criterios de buen estado

Define cómo se determina el estado: con cuánta frecuencia se verifica, cuánto tiempo se debe esperar una respuesta y cuántos intentos exitosos o con errores son decisivos.

Intervalo de verificación *

10

segundos

?

Tiempo de espera *

5

segundos

?

Umbral de buen estado *

3

resultados correctos consecutivos

?

Umbral de mal estado *

3

errores consecutivos

?

Al final damos click en GUARDAR y al final de la página nuevamente en GUARDAR.

Para verificar la reparación automática abrimos la terminal SSH de una de nuestras dos instancias, verificamos que en esa IP está funcionando la aplicación web y posteriormente matamos el proceso asociado con el programa java que ejecuta

nuestra aplicación web (mande una captura como la siguiente para comprobar este paso (**Item 4**):

```
ukraniocc@instance-group-ukranio-nf20:~$ ps -A | grep java
2023 ?          00:00:13 java
ukraniocc@instance-group-ukranio-nf20:~$ sudo kill -9 2023
```

salimos de la terminal SSH y verificamos en el navegador que nuestra aplicación ya no está corriendo, aunque si está activa la instancia. Sólo esperamos un tiempo para verificar que el reparador automático inicia el proceso de dar de baja la instancia considerada defectuosa y la reemplaza con una nueva.

IMPORTANTE: Al finalizar la práctica no olvide eliminar todos los recursos creados en Google Cloud como buckets, instancias y plantillas para evitar generar cargos en su cuenta bancaria.