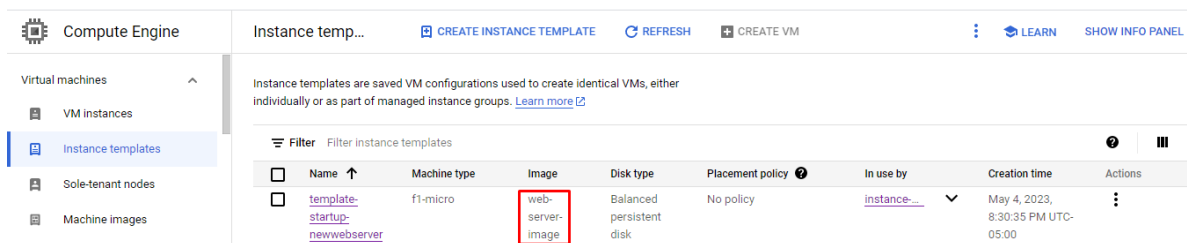


## Políticas de Auto-scaling y balanceador de carga.

A continuación se describe el proceso seguido para la implementación de políticas de autoescalping y el uso de un balanceador de carga en el despliegue de la solución.

De acuerdo con lo requerido para la entrega de la semana 5 se configuró un bucket en el servicio de almacenamiento de objetos para almacenar todos los archivos subidos por los usuarios, tanto los originales como los procesados. Se realizaron los cambios requeridos en el web server y el worker y posteriormente se generó una imagen del disco del web server.

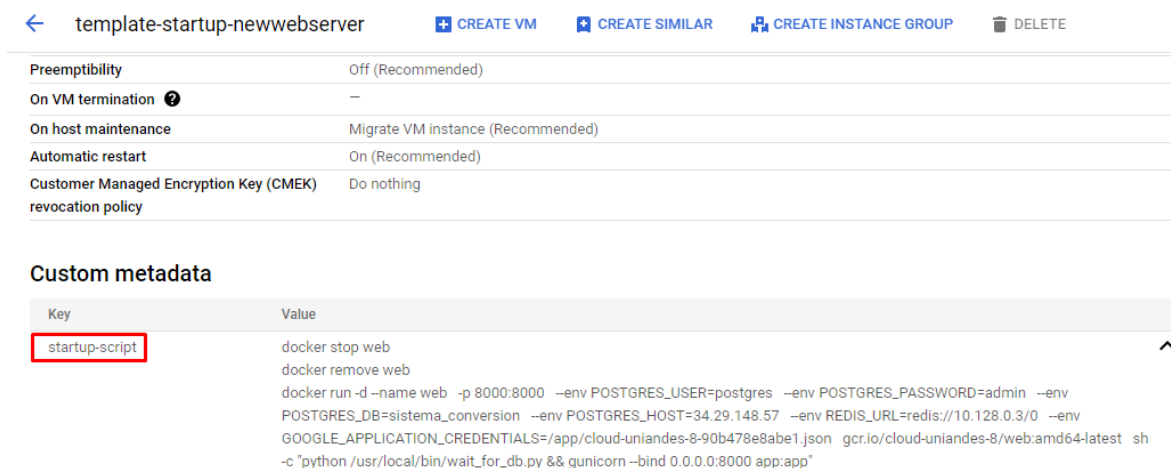
Usando el disco del web server creado previamente, se creó un instance template.



The screenshot shows the 'Instance templates' page in the Google Cloud Platform console. The left sidebar shows the navigation menu with 'Instance templates' selected. The main content area shows a table of instance templates. The first template, 'template-startup-newwebserver', is highlighted. The 'Image' column for this template is highlighted with a red box, showing 'web-server-image'.

Name	Machine type	Image	Disk type	Placement policy	In use by	Creation time	Actions
template-startup-newwebserver	f1-micro	web-server-image	Balanced persistent disk	No policy	instance-	May 4, 2023, 8:30:35 PM UTC-05:00	

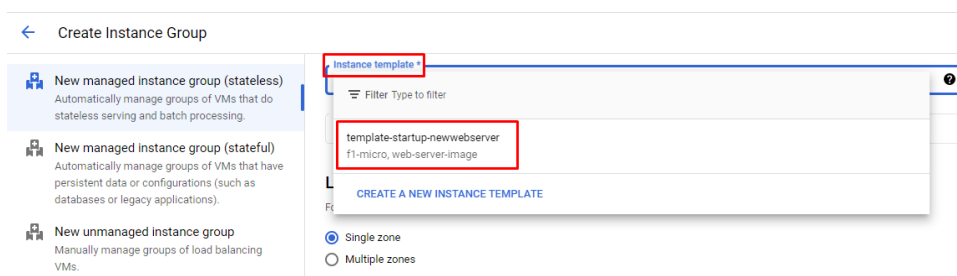
En este template se definió un startup script para correr la imagen del docker cargada en la instancia y garantizar la funcionalidad del web server al encender la máquina virtual.



The screenshot shows the 'Custom metadata' section for the 'template-startup-newwebserver' instance template. The 'startup-script' key is highlighted with a red box, showing a Docker-based startup script.

```
docker stop web
docker remove web
docker run -d --name web -p 8000:8000 --env POSTGRES_USER=postgres --env POSTGRES_PASSWORD=admin --env POSTGRES_DB=sistema_conversion --env POSTGRES_HOST=34.29.148.57 --env REDIS_URL=redis://10.128.0.3/0 --env GOOGLE_APPLICATION_CREDENTIALS=/app/cloud-unianandes-8-90b478e8abe1.json gcr.io/cloud-unianandes-8/web:amd64-latest sh -c "python /usr/local/bin/wait_for_db.py && gunicorn --bind 0.0.0.0:8000 app:app"
```

Posteriormente se creó un Instance Group en el cual se hace uso del Instance Template creado en el paso anterior.



The screenshot shows the 'Create Instance Group' page in the Google Cloud Platform console. The 'Instance template' dropdown is highlighted with a red box, showing 'template-startup-newwebserver'.

Instance template: template-startup-newwebserver  
f1-micro, web-server-image

Para la política de Autoscaling se definió que el número mínimo de instancias fuera 1 y el número máximo de instancias fuera 3 (de acuerdo a los requerimientos del entregable de la semana 5). El tipo

de señal utilizada para determinar cuándo escalar fue la utilización de CPU para la cual se definió un valor objetivo de 60%.

### Autoscaling

Use autoscaling to automatically add and remove instances to the group for periods of high and low load. [Learn more](#)

**Autoscaling mode**  
On: add and remove instances to the group

**Minimum number of instances \***  
1

**Maximum number of instances \***  
3

### Autoscaling signals

Use signals to help determine when to scale the group. [Learn more](#)

**CPU utilization: 60% (default)**  
Predictive autoscaling is off

[ADD A SIGNAL](#)

Se consideró un Cool down period 1200 segundos teniendo en cuenta el tiempo de inicialización de la aplicación para evitar que las políticas de autoescalado fueran aplicadas durante este tiempo.

### Cool down period

Specify how long it takes for your app to initialize from boot time until it is ready to serve. [Learn more](#)

**Cool down period \***  
1200

seconds ?

Para favorecer la disponibilidad del servicio, se incluyó una política de Autohealing basada en el health check que se muestra en la siguiente imagen.

### Autohealing

Autohealing recreates VM instances if your application cannot be reached by the health check. [Learn more](#)

**Health check**  
web-server-health-check (TCP)  
port: 8000, timeout: 5s, check interval: 10s, unhealthy threshold: 3 attempts

**Initial delay \***  
300

Seconds ?

En las configuraciones realizadas al Instance Group se incluyó el Port mapping para el puerto 8000.

### Port mapping




To send traffic to instance group through a named port, create a named port to map the incoming traffic to a specific port number, then go to "HTTP load balancing" to create a load balancer using this instance group.


**Port name 1**  
http



**Port numbers 1**  
8000

Finalmente, se creó el Load Balancing para el cual fue definido un servicio de backend con las siguientes especificaciones:

General properties

Load balancer type	Global External HTTP(S) Load Balancing (EXTERNAL_MANAGED)
Endpoint protocol	HTTP
In use by	<a href="#">http-lb</a>
Timeout 	120 seconds
Health check	<a href="#">web-server-health-check</a> <a href="#">VIEW HEALTH CHECK DETAILS</a>
Session affinity	None
Cloud CDN	Enabled <a href="#">VIEW CDN DETAILS</a>
Connection draining timeout	300 seconds
Custom request headers 	Currently there are no custom request headers configured
Custom response headers 	Currently there are no custom response headers configured
Logging	Disabled

Backends 

Name 	Type	Scope	Healthy	Autoscaling	Balancing mode
<a href="#">instance-group-startup-newwebserver</a>	Instance group	us-central1-a	 0 of 0	No configuration	Max backend

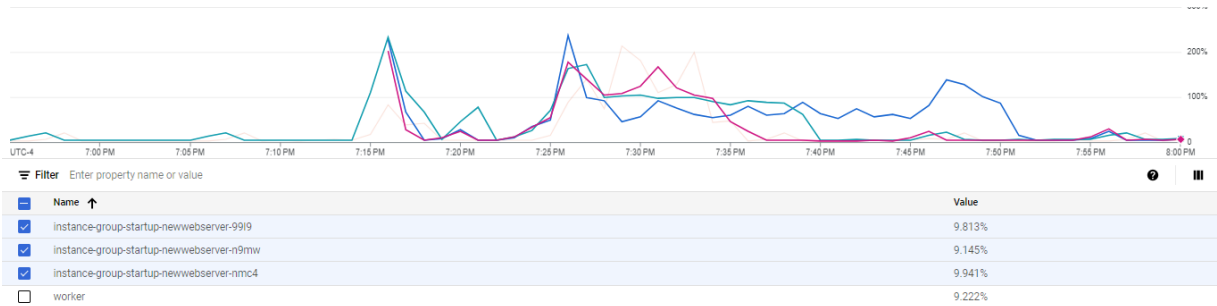
Las especificaciones del Frontend Service del Load Balancer son mostradas en la siguiente imagen:

http-lb-forwarding-rule	
Load balancing scheme	EXTERNAL_MANAGED
Network service tier	Premium
IP version	IPv4
External IP address	35.201.67.148:80
Protocol	TCP
Ports	80-80
Target	http-lb-target-proxy

Después de realizar esta configuración se realizaron las pruebas de los escenarios 1 y 2 apuntando a la IP externa del balanceador de carga por el puerto 80 donde se evidenció la ejecución de la política de Auto-scaling y el efecto del Load Balancing.



CPU Utilization by VM



CPU Utilization by Instance Group

