Universidad Rafael Landívar

Facultad de Ingeniería

Ingeniería Electrónica y Telecomunicaciones

Tecnologías de Virtualización y Data Centers

Lic. Juan Carlos Romero

**PROYECTO FINAL**

**INTEGRACIÓN DE NUBE DE NAUTILUS GROUP**

Juan Manuel Barillas – 1334816

Pablo David Flores

Javier Andrés Castañeda

Guatemala, 31 de mayo del 2024

ÍNDICE

[PLANTEAMIENTO DEL PROBLEMA 3](#_Toc167997418)

[1. SERVIDORES FRONT END: 4](#_Toc167997419)

[1.1 CONFIGURACIÓN APLICADA A LOS SERVIDORES A NIVEL DE TIPO DE MÁQUINAS. 4](#_Toc167997420)

[1.2 CÁLCULOS EFECTUADOS PARA ENTREGA DE VOLÚMENES RAID: 6](#_Toc167997421)

[1.3 PROCEDIMIENTO PARA CREACIÓN DE RAID SOLICITADO: 7](#_Toc167997422)

[1.4 MEDICIONES DE RENDIMIENTO DEL ARREGLO RAID 6, REALIZANDO UNA COMPARACIÓN CON UN SERVER CON ARREGLO RAID 5: 8](#_Toc167997423)

[1.5 ESTADÍSTICAS DE USO DE BALANCEADOR DE CARGA 9](#_Toc167997424)

[1.6 CAPTURAS DE IMPLEMENTACIÓN Y FUNCIONAMIENTO DE LA SOLUCIÓN: 10](#_Toc167997425)

[2. SERVIDORES DE BASE DE DATOS 11](#_Toc167997426)

[3. APLICATIVOS SERVERLESS 12](#_Toc167997427)

[3.1 SCRIPT DE DEPLOY DE SERVICIOS SERVERLESS 12](#_Toc167997428)

[3.2 ARQUITECTURA DE SERVICIOS WEB 14](#_Toc167997429)

[3.3 ESTADÍSTICAS DE CONSUMO DE LAS INSTANCIAS BALANCEADAS 15](#_Toc167997430)

[3.4 CAPTURAS DE IMPLEMENTACIÓN Y FUNCIONAMIENTO DE LA SOLUCIÓN. 16](#_Toc167997431)

[4. ANEXOS 17](#_Toc167997432)

[4.1 Pasos para desplegar HTML en Apache: 17](#_Toc167997433)

# PLANTEAMIENTO DEL PROBLEMA

La organización Nautilus Corp dedicada al transporte de carga marítima, ha contratado a tres ingenieros de la Universidad Rafael Landívar para llevar a cabo la migración de sus operaciones a la nube. La empresa se encuentra en una etapa crucial de la transformación digital y tiene como objetivo mejroar su eficiencia operativa, optimizar costos y garantizar una alta disponibilidad de sus servicios. Este cambio los llevará a modernizar su infraestructura tecnológica y a posicionarse como una entidad más competitiva y resiliente en el mercado global.

La decisión de migrar a la nube debe estar impulsada por factores estratégicos y operativos, conocidos como drivers, como la optimización de costos que les ayudará a reducir gastos en infraestructura física y mantenimiento. Escalabilidad, que les dará la capacidad de aumentar y mejorar sus recursos según la demanda, sin grandes inversiones iniciales. La alta disponibilidad, que asegurará que sus servicios siempre estén disponibles y la Seguridad, que ayudará a mejorar la seguridad de los datos y operaciones mediante servicios avanzados de protección de la nube.

Luego de analizados varios factores, se determina que la migración a la nube permitirá que Nautilus Corp logre modernizar su infraestructura tecnológica, mejorar su eficiencia operativa y ser más competitivos en el mercado global. Al adoptar las estrategias sugeridas como seguridad, estrategias centradas en la innovación y la escalabilidad, podrán ofrecer servicios de alta disponibilidad y mantenerse a la vanguardia del sector del transporte marítimo de carga. A continuación, se presentan las estrategias sugeridas:

# SERVIDORES FRONT END:

La implementación de los servidores front end para Nautilus Corp se es una parte crítica del proyecto para Nautilus Corp, especialmente considerando la necesidad de alta disponibilidad y optimización de costos. El objetivo principal de este apartado, es mostrar la documentación utilizada para desplegar los servidores web que puedan manejar el tráfico de manera eficiente y segura. Los pasos realizados para la crecaión de los servidores y el balanceo de cargas fueron los siguientes:

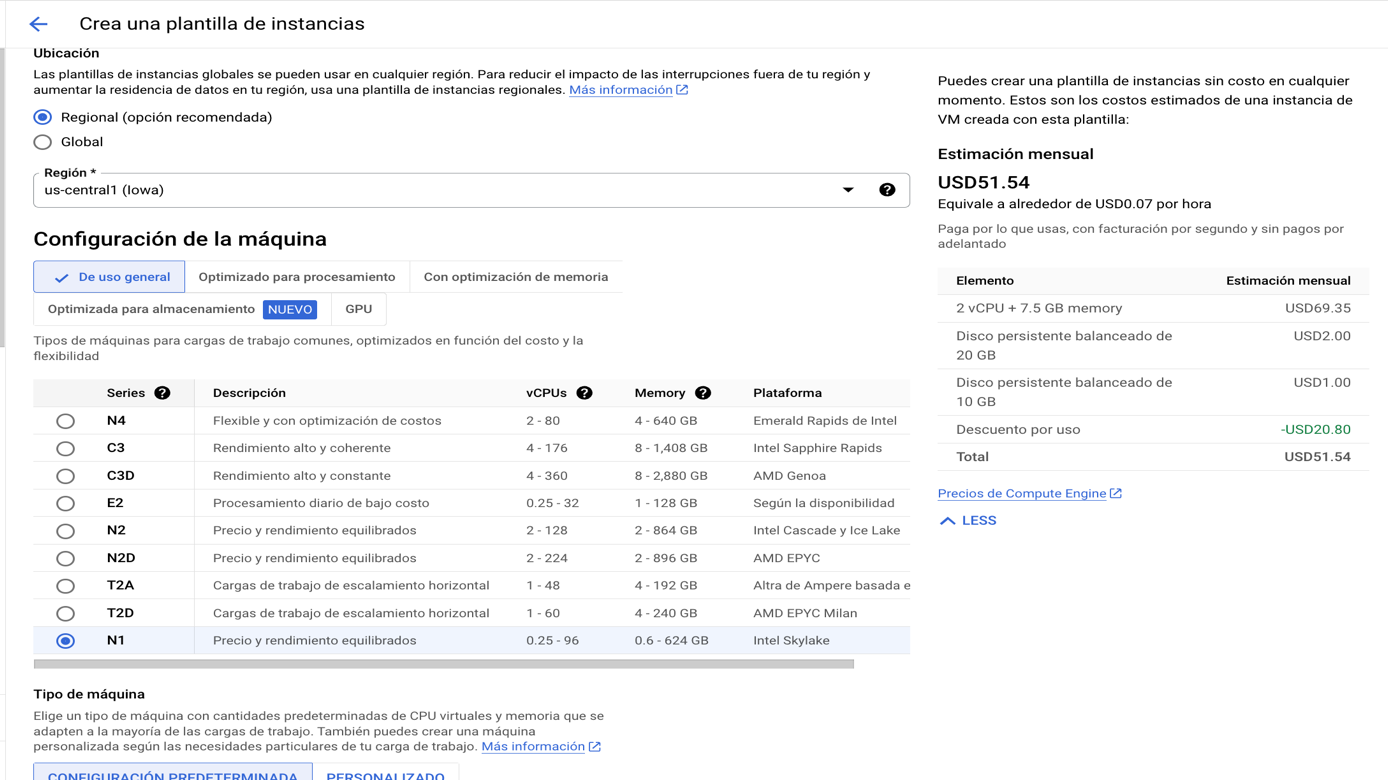
* Creación de la plantilla de operación de las VMs.
* Creación del grupo de instancias gestionadas sin estado (se gestiona automáticamente los grupos de VMs que realizan servicios sin estado).
* Creación de un estado de verificación , el cual determinará si las instancias de VM responden de forma apropiada al tráfico.
* Creación de una regla de Firewall para controlar el tráfico saliente y entrante de la instancia con rangos de IP.
* Reservación de la dirección IP externa.
* Creación del Balanceador de Cargas de aplicaciones que está orientado al público y es global.

A continuación, se detallan los pasos y configuraciones necesarias para configurar y desplegar los servidores de front end utilizando Google Cloud Platform para garantizar un rendimiento óptimo y una gestión eficiente.

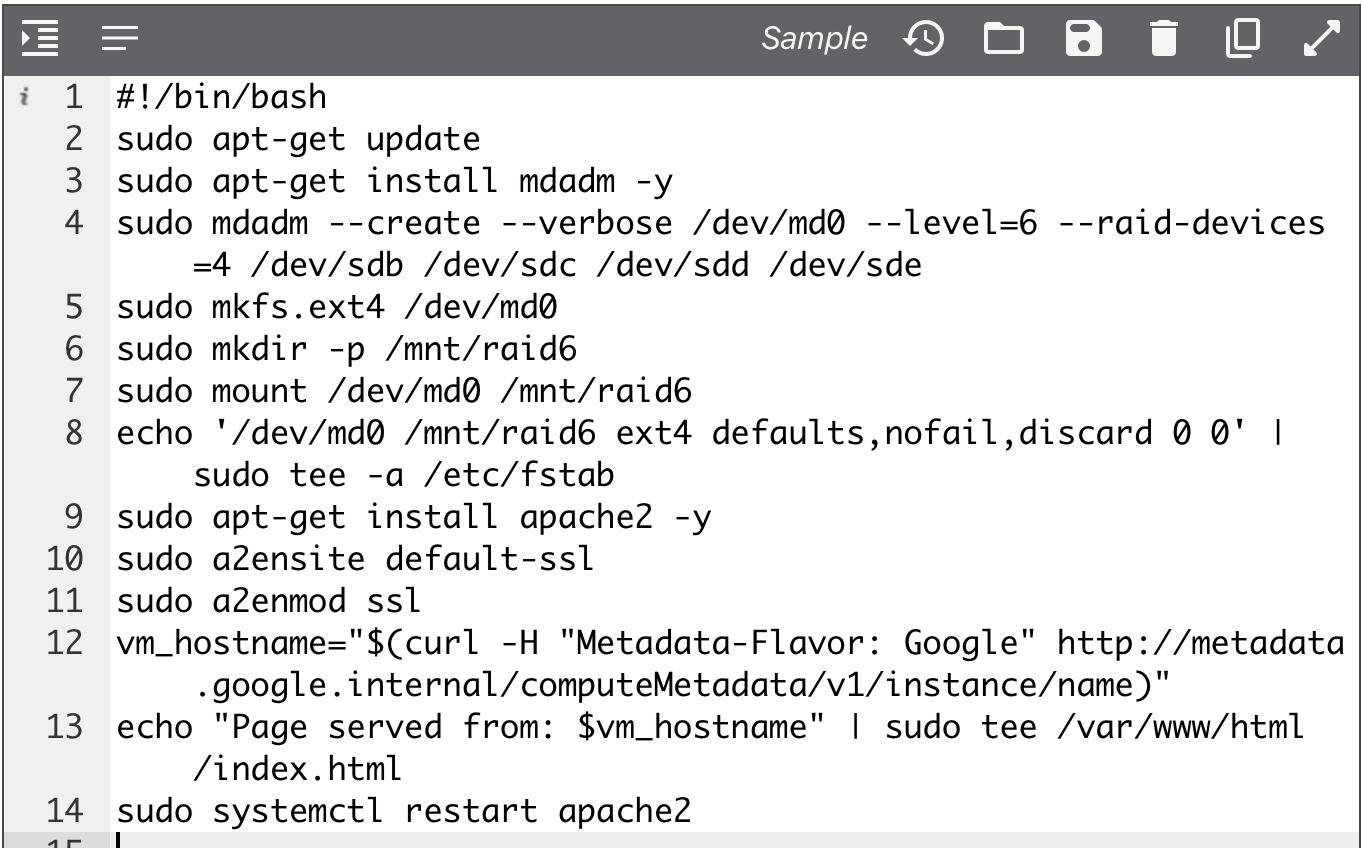
## CONFIGURACIÓN APLICADA A LOS SERVIDORES A NIVEL DE TIPO DE MÁQUINAS.

Para la implementación de los tres servidores web, los cuales están diseñados para poder alojar el portal web de la organización. Se utilizaron tres máquinas N1 standard 2, estas cuentan con dos CPU virtuales, 1 núcleo, 7.5GB de memoria y un disco de arranque de 10 GB con una imagen de Ubuntu 20.04 LTS previamente instalada con un costo estimado mensual de $51.54, equivalente a $0.07 por hora. A estos servidores web se puede acceder utilizando la dirección: ***http://34.36.145.197***

Las máquinas N1 fueron elegidas para los servidores web de Nautilus Corp debido a su capacidad para manejar tráfico web de manera eficiente con un equilibrio entre costo y rendimiento. La flexibilidad en la configuración de vCPU y memoria permite ajustar los recursos según las necesidades del tráfico. Además, se añadieron cuatro discos de 20 GB para el uso de una configuración de RAID 6, esto con el fin de proporcionar redundancia y mejorar la disponibilidad de los datos, garantizando que los servidores web puedan manejar tráfico de manera eficiente y segura.

****

**Imágen 1‑A: (Propia, 2024)** VMs a Utilizar para desplegar los Web Server.

Para la creación de dicha plantilla, se detalló un script de automatización para configurar el RAID 6 y la instalación de Apache 2, junto con la configuración necesaria. Este script de inicio automatiza la configuración del servidor al arrancar, asegurando que todos los servidores front end se configuren de manera uniforme y eficiente. Se incluyeron funciones para actualizar la lista de paquetes disponibles, instalar mdam, la cual es una herramienta para gestionar arreglos RAID en Linux, la configuración del arreglo RAID, instalación de un servidor web Apache, habilitar el módulo SSL de Apache y luego para reiniciarse para aplicar los cambios.

**Imágen 1‑B:** (2024, Propia) Script utilizado para la creación de las VMs.

Otra opción común en el mercado es optar por las máquinas E2, las cuales podrían ser una opción más rentable, sin embargo, estas no cuentan con la capacidad de poder alojar servidores web con cargas variables ya que cuenta con menos opciones de configuración y por lo general, únicamente son utilizadas en aplicaciones como servidores de prueba y de desarrollo, no de producción.

## CÁLCULOS EFECTUADOS PARA ENTREGA DE VOLÚMENES RAID:

Para la implementación de la infraestructura de servidores web, se establecieron requerimientos específicos en relación con la configuración de almacenamiento utilizando RAID 6, estos se diseñaron para garantizar la alta disponibilidad, integridad de los datos y eficiencia en el manejo del tráfico web. La ventaja de elegir un arreglo RAID 6 para este proyecto es que brinda alta disponibilidad de los datos, a través de la redundancia tolerando fallos sin interrumpir el servicio y garantizando la continuidad del negocio al permitir que los servidores se mantengan operativos y accesibles incluso durante fallas al disco. También, se tomó en cuenta la protección de datos ya que implementar este sistema permite que los datos estén protegidos contra la pérdida de datos mediante técnicas de redundancia y paridad. A continuación, se presentan los cálculos efectuados para entregar un volúmen útil de almacenamiento total de 40 GB:

(1)

Como se observa en los cálculos, el volúmen útil disponible en el arreglo RAID 6 es de 40 GB, esta es la capacidad total de almacenamiento que puede ser utilizada para guardar datos, mientras que los otros dos discos proporcionan la redundancia necesaria para asegurar la protección de los datos a través de bits de paridad.

## PROCEDIMIENTO PARA CREACIÓN DE RAID SOLICITADO:

La solicitud de la creación de un RAID 6 en la solución de Nautilus Corp se basa en factores que buscan asegurar la integridad, disponibilidad y rendimiento de los datos en los servidores web. A continuación, se detalla el procedimiento seguido para la creación del Raid 6, garantizando la alta disponibilidad y la alta tolerancia a fallos.

Debido a que la plantilla de automatización con las que fueron creadas las instancias utilizadas para el servidor web ya realizaban la configuración básica del servidor como ingresar al modo privilegiado, actualizar el sistema, instalar mdam para la creación del RAID, se omiten estos pasos:

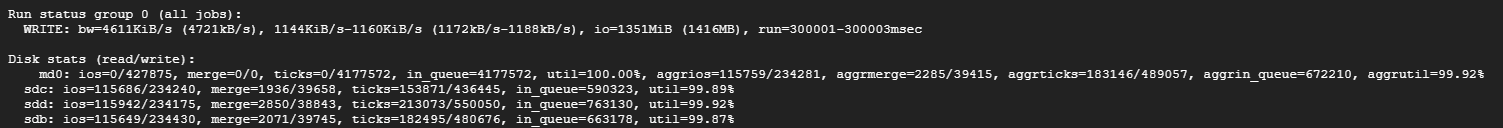
1. Verificar los volúmenes de los discos para confirmar que si se cuente con el almacenamiento deseado a través del comando cat /proc /partitions.
2. Preparar los discos para el raid, en donde para cada uno de los discos (sdb, sdc, sdd y sde) se debe de crear una partición de Linux Raid Autodetect utilizando *fdisk*.
3. Actualizar la tabla de particiones utilizando el comando *partprobe.*
4. Crear el Arreglo RAID 6 ejecutando el siguiente comando: *mdadm --create /dev/md0 --level=6 --raid-devices=4 /dev/sdb1 /dev/sdc1 /dev/sdd1 /dev/sde1*. Este comando crea el arreglo RAID como tal en donde el nombre asignado es md0, se indica que se contará con 4 dispositivos para el RAID de nivel 6 y los discos utilizados para el arreglo.
5. Luego de completado la creación el arreglo, se debe de formatear el volumen RAID por medio del comando mkfs.ext4/dev/md0, con el cual se podrá almacenar información.

## MEDICIONES DE RENDIMIENTO DEL ARREGLO RAID 6, REALIZANDO UNA COMPARACIÓN CON UN SERVER CON ARREGLO RAID 5:

Para la solución presentada, se evaluaron factores como los costos de ambas implementaciones y la criticidad de la disponibilidad y redundancia de datos. En este apartado, se presenta una comparación del rendimiento del arreglo RAID 6 utilizado en la implementación de Nautilus Corp:

**Server RAID 5:**

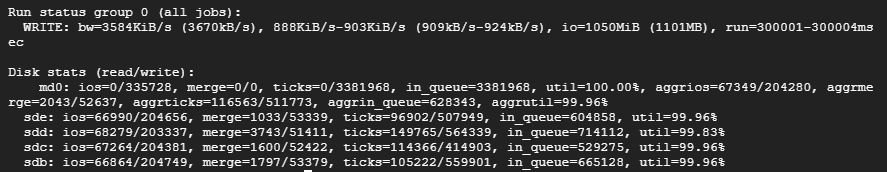
Se puede observar una velocidad de escritura de 4,611 KiB/s es decir, una velocidad de escritura promedio de 4721 kB/s, un tamaño de entrada y salida (I/O) de 1351 MiB, el cual se refiere a la cantidad de datos que se transfieren en una sola operación de lectura o escritura entre la memoria y el almacenamiento. Este presenta un tamaño de I/O mayor y una velocidad de escritura más alta, lo que puede ser beneficioso para aplicaciones que requieren un rendimiento de escritura más rápido.



**Imágen 1‑C:** (Propia, 2024) Velocidad de Escritura de un Arreglo RAID 5.

**Server RAID 6:**

Cuenta con una velocidad de escritura de 3,585 KiB/S, es decir, una velocidad de escritura promedio de 3,670 kB/s y un tamaño de I/O de 1050. Aunque tiene un tamaño de I/O menor y una velocidad de escritura más baja, ofrece mayor redundancia y tolerancia a fallos, lo que es crítico para la seguridad y disponibilidad de los datos.

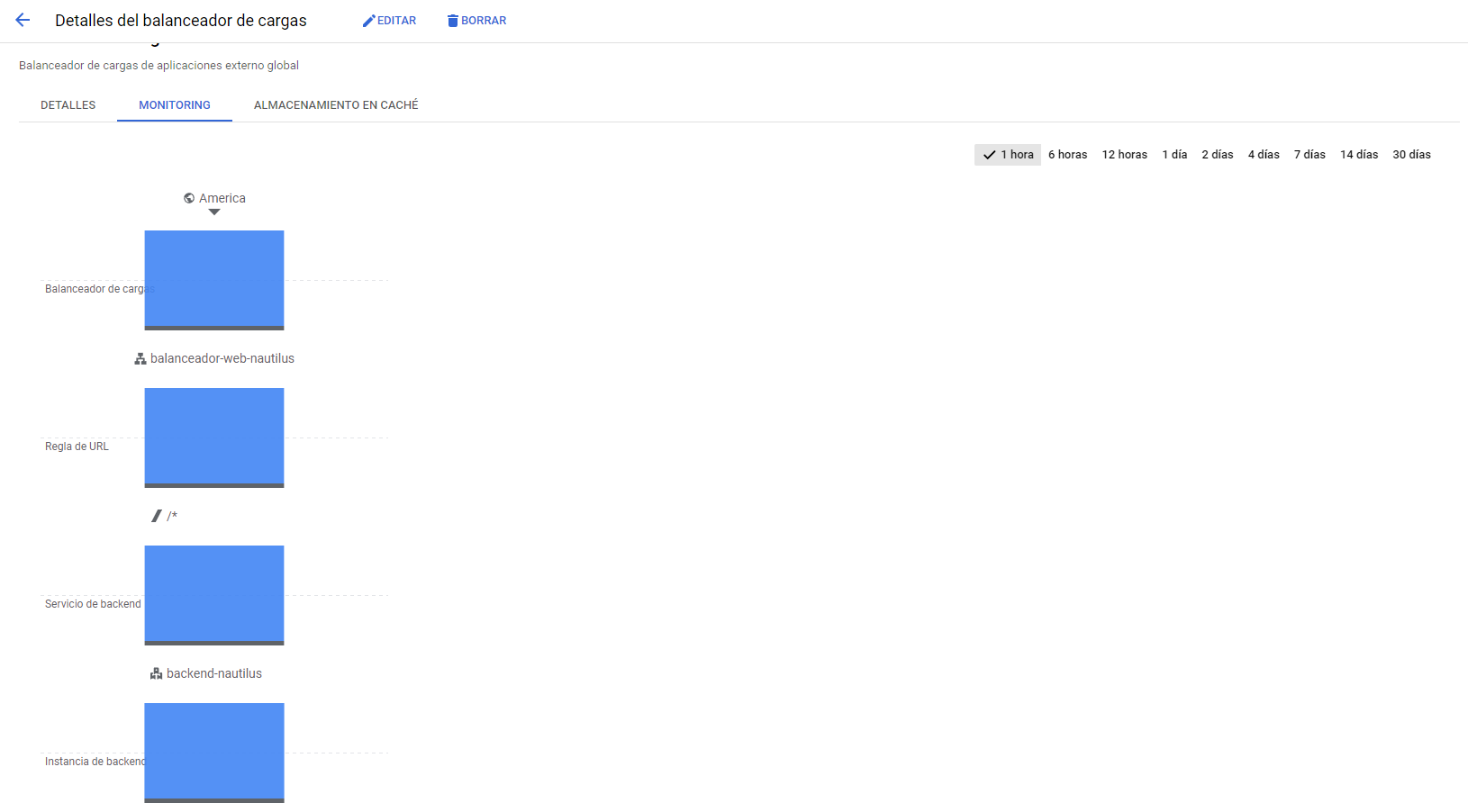


**Imágen 1‑D:** (Propia, 2024) Se muestra la Velocidad de Escritura de un Arreglo.

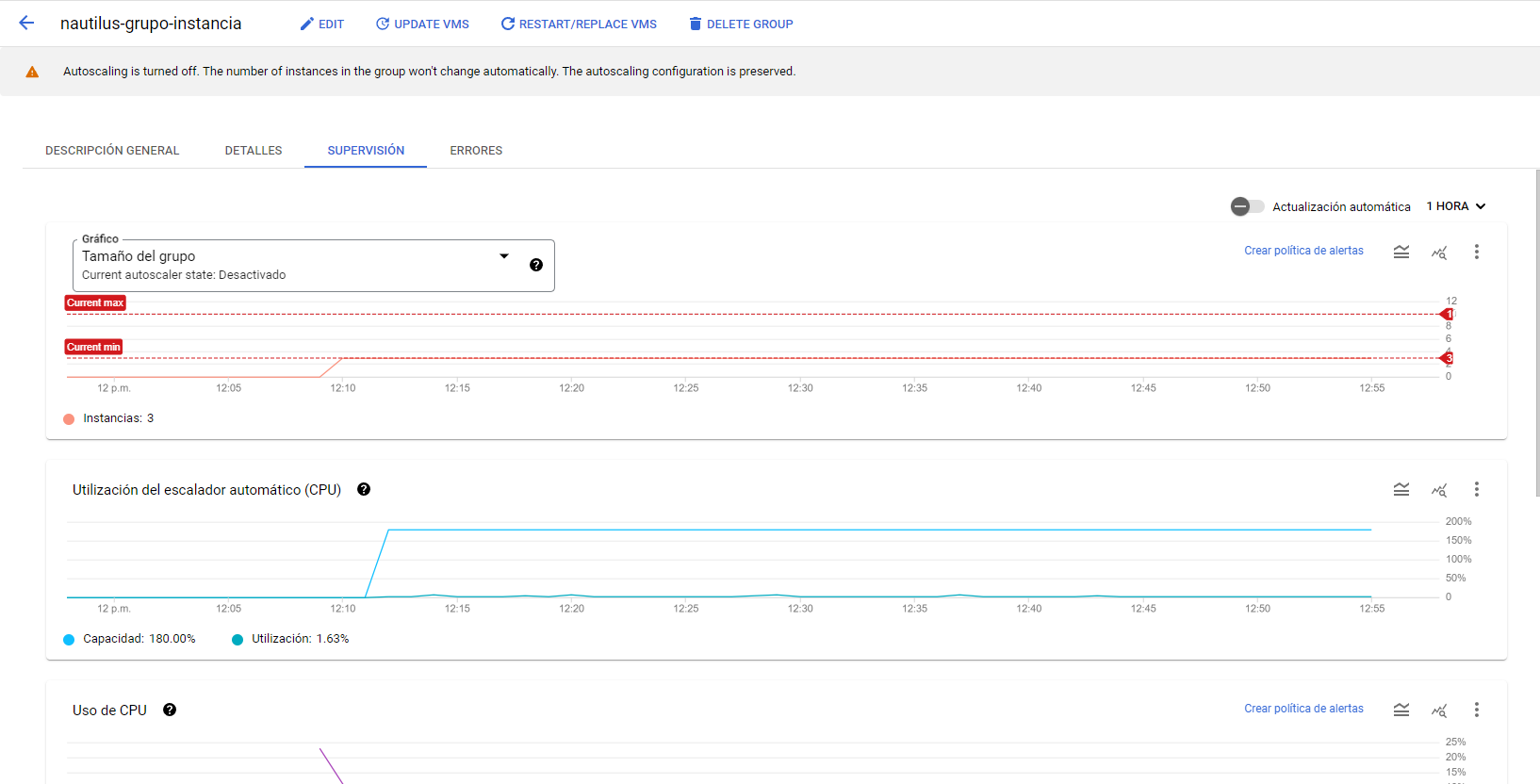
Para Nautilus Corp, es importante contar con alta disponibilidad, redundancia de sus datos y brindar protección a sus datos, es por eso que se considera la opción de RAID 6 aunque tenga un tamaño de escritura y lectura menor y una velocidad de escritura más baja, ya que la prioridad es brindar disponibilidad. RAID 5 es adecuado para aplicaciones donde el rendimiento de escritura es más crítico y se puede tolerar una menor redundancia de datos. Aunque RAID 6 implique un mayor costo inicial debido a la necesidad de más discos de paridad, su mayor tolerancia a fallos y protección contra pérdida de datos hacen que esta sea la elección correcta para servidores web críticos lo que puede resultar en ahorro d ecostos a largo plazo y reducir el riesgo de inactividad de estos servidores.

## ESTADÍSTICAS DE USO DE BALANCEADOR DE CARGA

A continuación, se muestran las estadísticas del uso del balanceador de cargas en Google Cloud Platform para Nautilus Corp. En estas se muestra la carga y el tráfico manejado por el balanceador de cargas, así como su distribución a través de los diferentes servicios de backend creados. Estas estadísticas ayudan a tener una visión clara de la eficiencia del balanceador permitiendo tomar decisiones en caso se necesite optimizar o aumentar el poder de cómputo de la infraestructura.

****

**Imágen 1‑E:** (Propia, 2024) Se muestran los detalles y estadísticas del balanceador de carga.

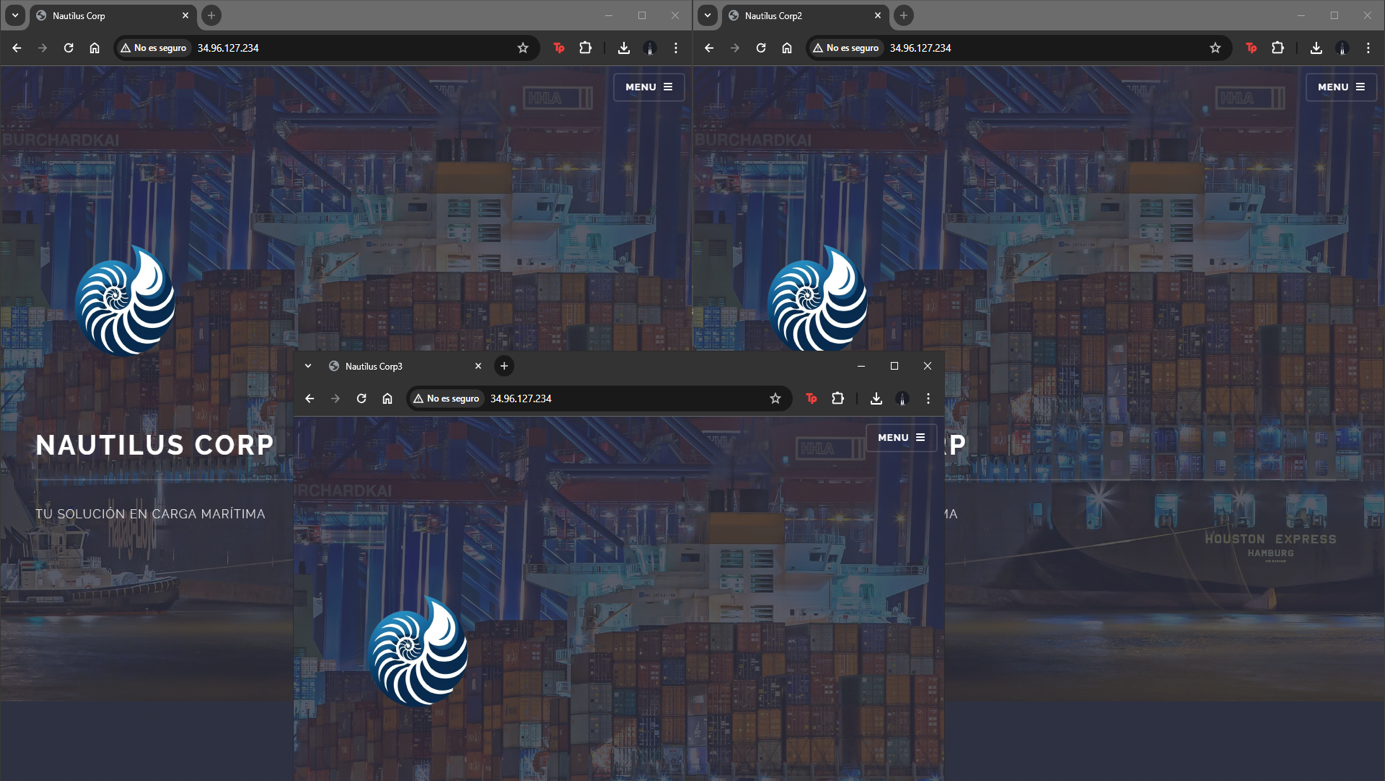
****

**Imágen 1‑F:** (Propia, 2024) Se muestran los detalles y estadísticas del uso del balanceador de cargas.

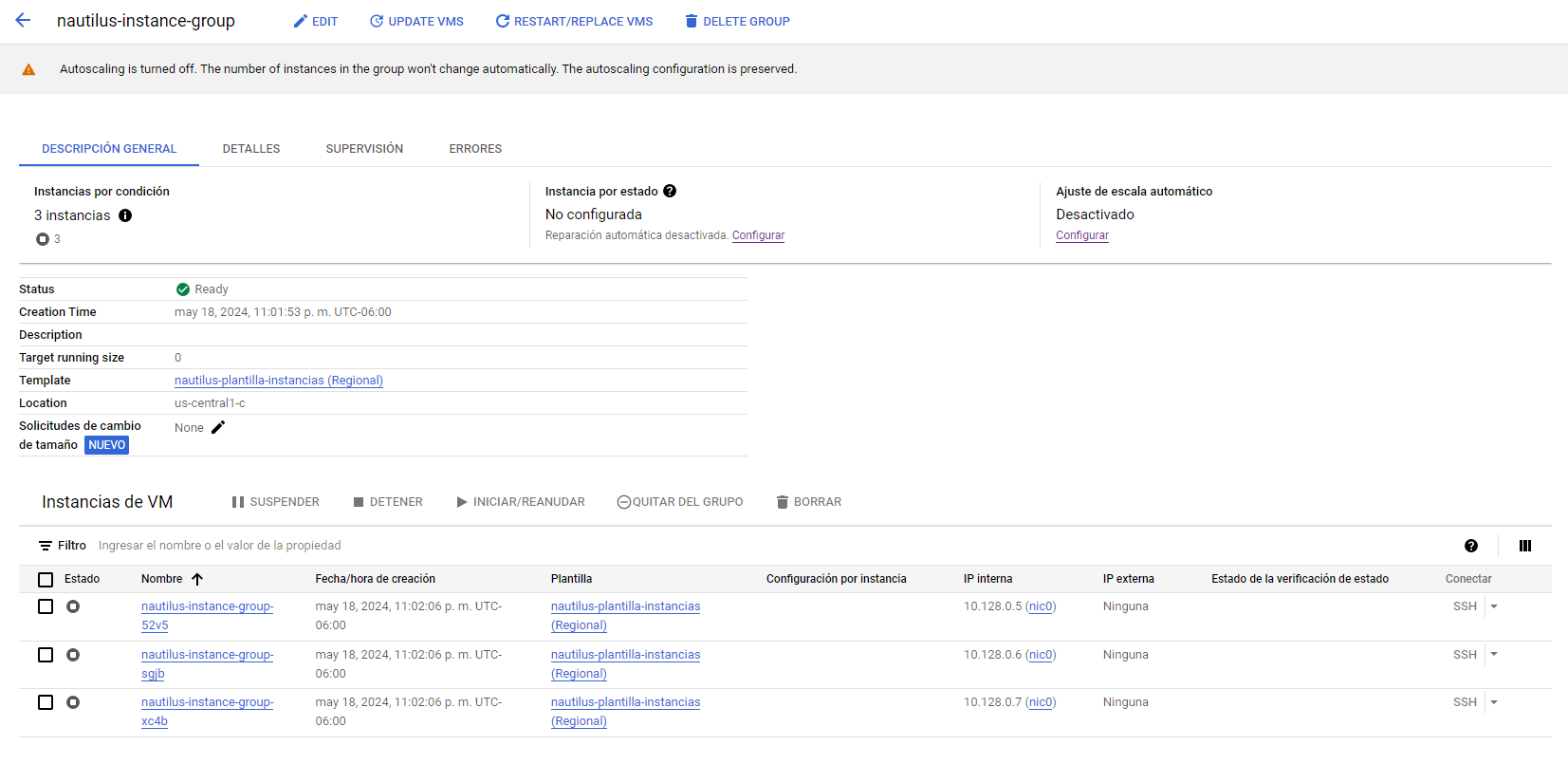
## CAPTURAS DE IMPLEMENTACIÓN Y FUNCIONAMIENTO DE LA SOLUCIÓN:

Para la solución presentada, se implementó un balanceador de cargas de aplicaciones global y orientado al público en GCP, este es crucial para distribuir el tráfico entrante de manera eficiente, asegurando alta disponibilidad y rendimiento.

El balanceador de cargas utilizado permite distribuir el tráfico de red entrante a múltiples instancias de servidores web, optimizando el uso de recursos y garantizando la disponibilidad, al implementarlo de manera global, el tráfico se puede gestionar a nivel mundial, proporcionando una única dirección IP pública que simplifica el acceso y la gestión del servicio. El uso de un balanceador de carga global permite que todas las instancias de servidores web compartan la misma dirección IP pública lo que se logra mediante asignar una dirección IP estática global al balanceador de cargas, la cual es utilizada para todas las solicitudes entrantes, el balanceador de carga distribuye el tráfico entrante a las instancias backend de manera equitatitva y eficiente, permitiendo así que todas las instancias puedan recibir tráfico de manera equilibrada , mejorando la resiliencia y la escalabilidad del servicio web. A continuación los resultados:

****

**Imágen 1‑G:** (Propia, 2024) Se muestran las capturas del funcionamiento de la solución para los servidores web.



**Imágen 1‑H:** (Propia, 2024) Se muestran las capturas de la implementación de la solución para los servidores web.

# SERVIDORES DE BASE DE DATOS

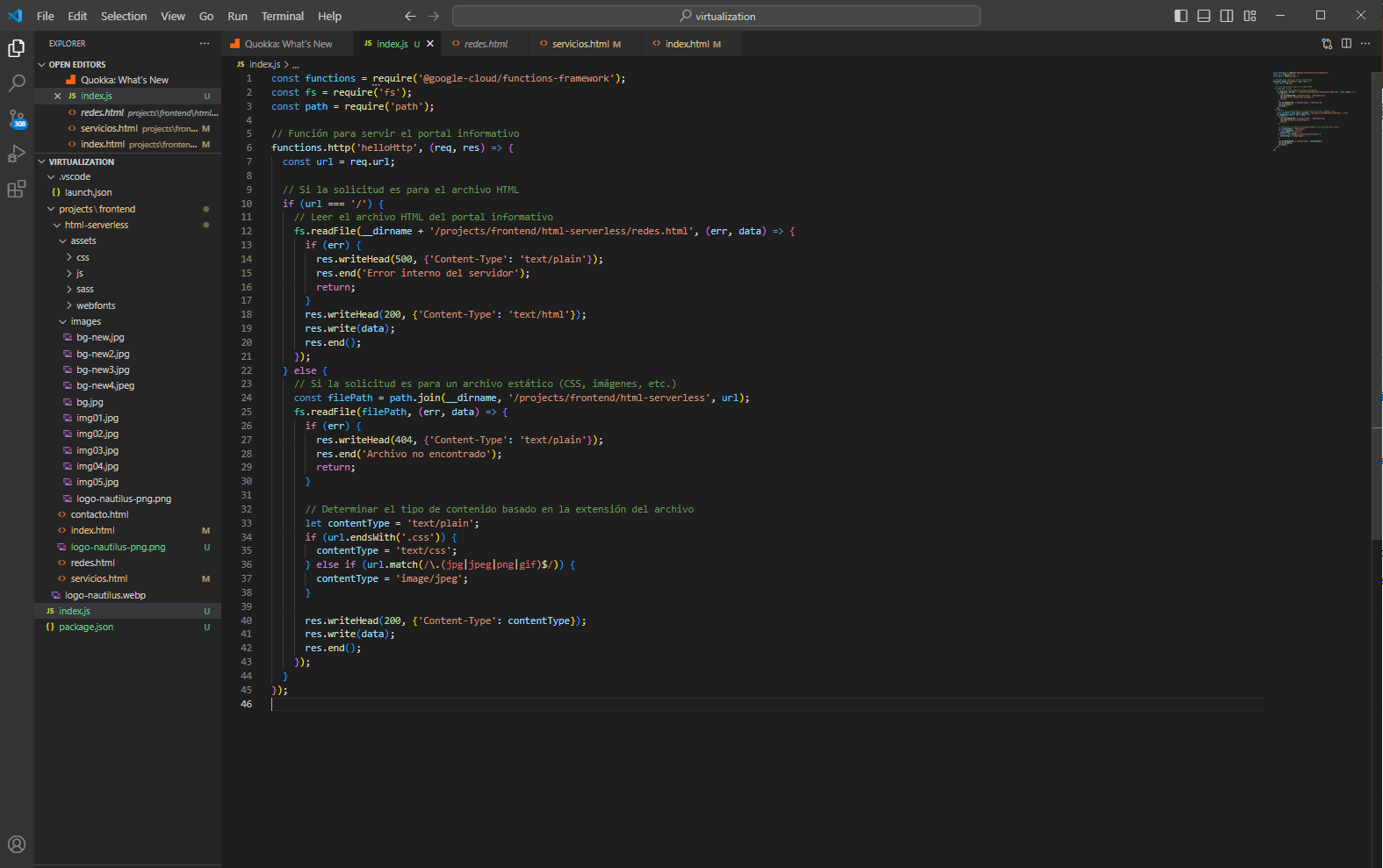
# APLICATIVOS SERVERLESS

La adopción de aplicativos serverless proporciona a Nautilus Corp. una forma eficiente y escalable de gestionar sus aplicaciones y servicios web, especialmente con el objetivo que tiene la compañía de disminuir sus costos. El objetivo de este apartado es mostrar los pasos realizados para realizar el despliegue de tres portales informativos HTML, cada uno de ellos manejado por un servicio serverless. Todos los servicios desplegados cuentan con dos versiones sobre las cuales se puedan balancear los requerimientos realizados por parte de los clientes en una medida de 50% para cada instancia de forma aleatoria.

Los aplicativos serverless son aplicaciones que se ejecutan en entornos en la nube sin necesidad de gestionar servidores, en lugar de estos, se utilizan servicios en la nube que ejecutan y gestionan el código de la aplicación en respuesta a eventos específicos. Su funcionalidad está basada en eventos y cada evento desencadena una función específica que se ejecuta automáticamente. La adopción de aplicativos serverless proporciona a Nautilus Corp una forma eficiente y escalable de gestionar sus aplicaciones y servicios web. Brindando beneficios como la automatización, escalabilidad automática y la reducción de costos, por lo que esta arquitectura es una opción beneficiosa para soportar el crecimiento y las operaciones digitales de la compañía, asegurando una implementación robusta y flexible que se adapte a sus necesidades presentes y futuras.

## SCRIPT DE DEPLOY DE SERVICIOS SERVERLESS

Para desplegar los servicios serverless de la compañía, se creó un script utilizando Google Cloud Functions para servir archivos estáticos y responder a las solicitudes HTTP. Este script permite servir un archivo HTML principal desde la raíz del sitio web y manejar solicitudes para archivos estáticos como CSS y las imágenes. Primero, se realiza el despliegue de serverless a través de la función de Google Cloud Functions, eliminando la necesidad de gestionar servidores. Luego, se genera un manejo de solicitudes HTTP, la cual responde a solicitudes HTTP entrantes y sirve archivos estáticos directamente desde el almacenamiento de la nube asegurando que el contenido del sitio web sea accesible para los usuarios finales. A continuación, se presenta el script generado para el servicio de redes sociales:



**Imágen 3‑A:** (Propia, 2024) Se muestra el script de deploy de los servicios serverless.

## ARQUITECTURA DE SERVICIOS WEB

## ESTADÍSTICAS DE CONSUMO DE LAS INSTANCIAS BALANCEADAS

## CAPTURAS DE IMPLEMENTACIÓN Y FUNCIONAMIENTO DE LA SOLUCIÓN.

# ANEXOS

## PASOS PARA DESPLEGAR HTML EN APACHE WEB SERVER:

1. Clonar el repositorio con sudo cd /var/www/html/
2. Sudo git clone link de repositorio
3. rm index.html
4. Buscar y copiar el path dentro del repositorio que haga referencia al index.html
5. Modificar el archivo de configuración de apache y copiar la ruta:
6. sudo nano /etc/apache2/sites-available/000-default.conf
7. copiar la ruta en DocumentRoot
8. reinciar el server: systemctl restart apache2

Nota: no es necesario habilitar permisos ya que la carpeta ya está ubicada en el directorio.

/var/www/html/virtualization/projects/frontend/proyecto-dc-html2/

/var/www/html/virtualization/projects/frontend/proyecto-dc-html3/

/var/www/html/virtualization/projects/frontend/proyecto-dc-html