

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 - 1164720
Javier Andrés Castañeda - 1290520

Guatemala, 31 de mayo del 2024

ÍNDICE

ÍNDICE	2
PLANTEAMIENTO DEL PROBLEMA	4
1. SERVIDORES FRONTEND:	5
1.1 CONFIGURACIÓN APLICADA A LOS SERVIDORES A NIVEL DE TIPO DE MÁQUINAS.	5
1.2 CÁLCULOS EFECTUADOS PARA ENTREGA DE VOLÚMENES RAID:	7
1.3 PROCEDIMIENTO PARA CREACIÓN DE RAID SOLICITADO:	8
1.4 MEDICIONES DE RENDIMIENTO DEL ARREGLO RAID 6, REALIZANDO UNA COMPARACIÓN CON UN SERVER CON ARREGLO RAID 5:	9
1.5 ESTADÍSTICAS DE USO DE BALANCEADOR DE CARGA	10
1.6 CAPTURAS DE IMPLEMENTACIÓN Y FUNCIONAMIENTO DE LA SOLUCIÓN:	11
2. SERVIDORES DE BASE DE DATOS	13
2.1 CONFIGURACIÓN APLICADA A LOS SERVIDORES A NIVEL DE TIPO DE MÁQUINAS, DIMENSIONAMIENTO DE DISCOS DUROS, MEMORIA RAM Y PROCESADORES:	13
2.2 CONFIGURACIÓN DEL FILESTORE Y SCRIPT DE ASOCIACIÓN EN SISTEMA OPERATIVO.	15
2.3 JUSTIFICACIÓN DE LA DECISIÓN EN CUANTO AL USO DE LA BASE DE DATOS A DESPLEGAR EN EL FILESTORE.	17
2.4 CAPTURAS DE IMPLEMENTACIÓN Y FUNCIONAMIENTO DE LA SOLUCIÓN.	18
3. APLICATIVOS SERVERLESS	19
3.1 SCRIPT DE DEPLOY DE SERVICIOS SERVERLESS	19
3.2 ARQUITECTURA DE SERVICIOS WEB	21
3.3 ESTADÍSTICAS DE CONSUMO DE LAS INSTANCIAS BALANCEADAS	22
3.4 CAPTURAS DE IMPLEMENTACIÓN Y FUNCIONAMIENTO DE LA SOLUCIÓN.	22
4. SERVIDORES DE ALMACENAMIENTO – BUCKETS	25
4.1 TAXONOMÍA DE LOS ARBOLES DE ARCHIVOS DE AMBOS BUCKETS	25
4.2 POLÍTICAS DE ACCESO – JUSTIFICACIÓN DE ROLES	26
4.3 CAPTURAS DE IMPLEMENTACIÓN Y FUNCIONAMIENTO DE LA SOLUCIÓN	30
5. SERVICIOS DE BACKUP DE BD	31
5.1 PLAN DE BACKUP	31
5.2 CAPTURAS DE IMPLEMENTACIÓN Y FUNCIONAMIENTO DE LA SOLUCIÓN.	32
6. CONTENEDORES	33

6.1	SCRIPT DE IMPLEMENTACIÓN DEL CONTENEDOR	33
6.2	CAPTURA DE CONFIGURACIÓN DE MÁQUINA DE IMPLEMENTACIÓN	33
6.3	CAPTURAS DE IMPLEMENTACIÓN EN LA NUBE.	34
6.4	CAPTURAS DE IMPLEMENTACIÓN Y FUNCIONAMIENTO DE LA SOLUCIÓN	34
ANEXOS		35
a.	PASOS PARA DESPLEGAR HTML EN APACHE WEB SERVER:	35

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 mejorar 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:

1. SERVIDORES FRONTEND:

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 creación 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.

1.1 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.

Crea una plantilla de instancias

Ubicación

Las plantillas de instancias globales se pueden usar en cualquier región. Para reducir el impacto de las interrupciones fuera de tu región y aumentar la residencia de datos en tu región, usa una plantilla de instancias regionales. [Más información](#)

☒

Regional (opción recomendada)

☐

Global

Región *

us-central1 (Iowa)

Configuración de la máquina

☒ De uso general

Optimizado para procesamiento

Con optimización de memoria

Optimizada para almacenamiento

NUEVO

GPU

Tipos de máquinas para cargas de trabajo comunes, optimizados en función del costo y la flexibilidad

Serie ?	Descripción	vCPUs ?	Memory ?	Plataforma
<input type="radio"/> N4	Flexible y con optimización de costos	2 - 80	4 - 640 GB	Emerald Rapids de Intel
<input type="radio"/> C3	Rendimiento alto y coherente	4 - 176	8 - 1,408 GB	Intel Sapphire Rapids
<input type="radio"/> C3D	Rendimiento alto y constante	4 - 360	8 - 2,880 GB	AMD Genoa
<input type="radio"/> E2	Procesamiento diario de bajo costo	0.25 - 32	1 - 128 GB	Según la disponibilidad
<input type="radio"/> N2	Precio y rendimiento equilibrados	2 - 128	2 - 864 GB	Intel Cascade y Ice Lake
<input type="radio"/> N2D	Precio y rendimiento equilibrados	2 - 224	2 - 896 GB	AMD EPYC
<input type="radio"/> T2A	Cargas de trabajo de escalamiento horizontal	1 - 48	4 - 192 GB	Altra de Ampere basada e
<input type="radio"/> T2D	Cargas de trabajo de escalamiento horizontal	1 - 60	4 - 240 GB	AMD EPYC Milan
<input checked="" type="radio"/> N1	Precio y rendimiento equilibrados	0.25 - 96	0.6 - 624 GB	Intel Skylake

Tipo de máquina

Elige un tipo de máquina con cantidades predeterminadas de CPU virtuales y memoria que se adapten a la mayoría de las cargas de trabajo. También puedes crear una máquina personalizada según las necesidades particulares de tu carga de trabajo. [Más información](#)

CONFIGURACIÓN PREDETERMINADA

PERSONALIZADO

Puedes crear una plantilla de instancias sin costo en cualquier momento. Estos son los costos estimados de una instancia de VM creada con esta plantilla:

Estimación mensual

USD51.54

Equivale a alrededor de USD0.07 por hora

Paga por lo que usas, con facturación por segundo y sin pagos por adelantado

Elemento	Estimación mensual
2 vCPU + 7.5 GB memory	USD69.35
Disco persistente balanceado de 20 GB	USD2.00
Disco persistente balanceado de 10 GB	USD1.00
Descuento por uso	-USD20.80
Total	USD51.54

[Precios de Compute Engine](#)

^

 LESS

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 frontend se configuren de manera uniforme y eficiente. Se incluyeron funciones para actualizar la lista de paquetes disponibles, instalar mdadm, 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.

```

1 #!/bin/bash
2 sudo apt-get update
3 sudo apt-get install mdadm -y
4 sudo mdadm --create --verbose /dev/md0 --level=6 --raid-devices
   =4 /dev/sdb /dev/sdc /dev/sdd /dev/sde
5 sudo mkfs.ext4 /dev/md0
6 sudo mkdir -p /mnt/raid6
7 sudo mount /dev/md0 /mnt/raid6
8 echo '/dev/md0 /mnt/raid6 ext4 defaults,nofail,discard 0 0' |
   sudo tee -a /etc/fstab
9 sudo apt-get install apache2 -y
10 sudo a2ensite default-ssl
11 sudo a2enmod ssl
12 vm_hostname="$(curl -H "Metadata-Flavor: Google" http://metadata
   .google.internal/computeMetadata/v1/instance/name)"
13 echo "Page served from: $vm_hostname" | sudo tee /var/www/html
   /index.html

```

14 **Imagen 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.

1.2 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 volumen útil de almacenamiento total de 40 GB:

$$CT = (N - 2) * C$$

$$CT = (4 - 2) * 20 \text{ GB}$$

$$CT = 40 \text{ GB}$$

(1)

Como se observa en los cálculos, el volumen 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.

1.3 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 mdadm para la creación del RAID, se omiten estos pasos:

1. Verificar los volúmenes de los discos para confirmar que si se cuenta 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.

1.4 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.

```
Run status group 0 (all jobs):  
WRITE: bw=4611KiB/s (4721kB/s), 1144KiB/s-1160KiB/s (1172kB/s-1188kB/s), io=1351MiB (1416MB), run=300001-300003msec  
  
Disk stats (read/write):  
md0: ios=0/427875, merge=0/0, ticks=0/4177572, in_queue=4177572, util=100.00%, aggrrios=115759/234281, aggrmerge=2285/39415, aggrticks=183146/489057, aggrin_queue=672210, aggrutil=99.92%  
sdc: ios=115686/234240, merge=1936/39658, ticks=153871/436445, in_queue=590323, util=99.89%  
sdd: ios=115942/234175, merge=2850/38843, ticks=213073/550050, in_queue=763130, util=99.92%  
sdb: ios=115649/234430, merge=2071/39745, ticks=182495/480676, in_queue=663178, util=99.87%
```

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.

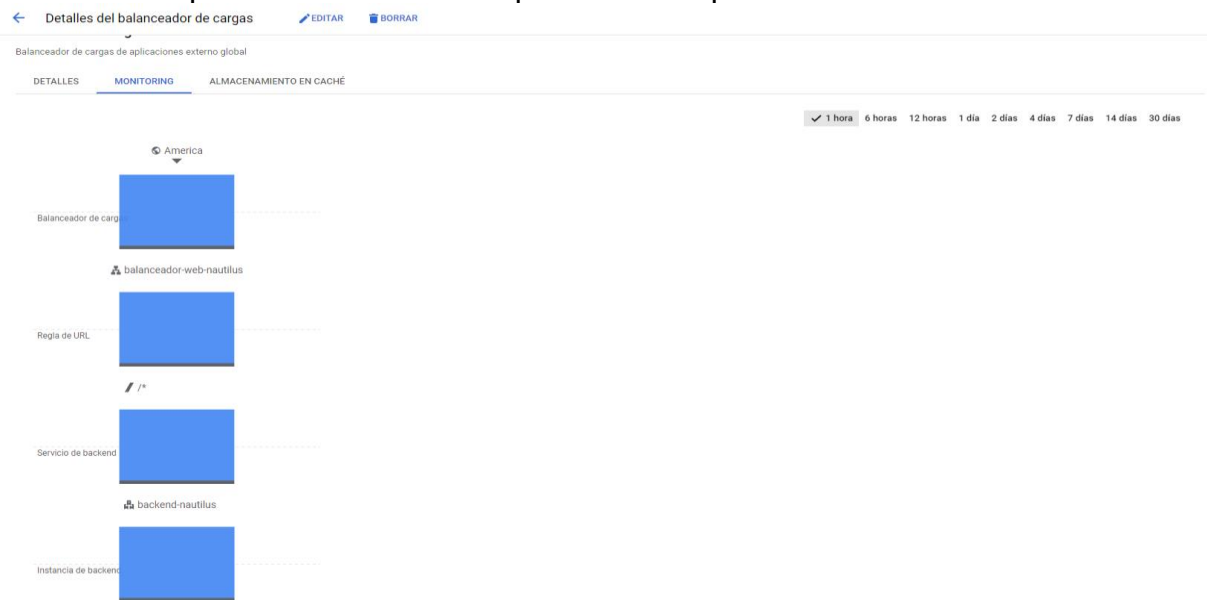
```
Run status group 0 (all jobs):  
WRITE: bw=3584KiB/s (3670kB/s), 888KiB/s-903KiB/s (909kB/s-924kB/s), io=1050MiB (1101MB), run=300001-300004msec  
  
Disk stats (read/write):  
md0: ios=0/335728, merge=0/0, ticks=0/3381968, in_queue=3381968, util=100.00%, aggrrios=67349/204280, aggrmerge=2043/52637, aggrticks=116563/511773, aggrin_queue=628343, aggrutil=99.96%  
sde: ios=66990/204656, merge=1033/53339, ticks=96902/507949, in_queue=604858, util=99.96%  
sdd: ios=68279/203337, merge=3743/51411, ticks=149765/564339, in_queue=714112, util=99.83%  
sdc: ios=67264/204381, merge=1600/52422, ticks=114366/414903, in_queue=529275, util=99.96%  
sdb: ios=66864/204749, merge=1797/53379, ticks=105222/559901, in_queue=665128, util=99.96%
```

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

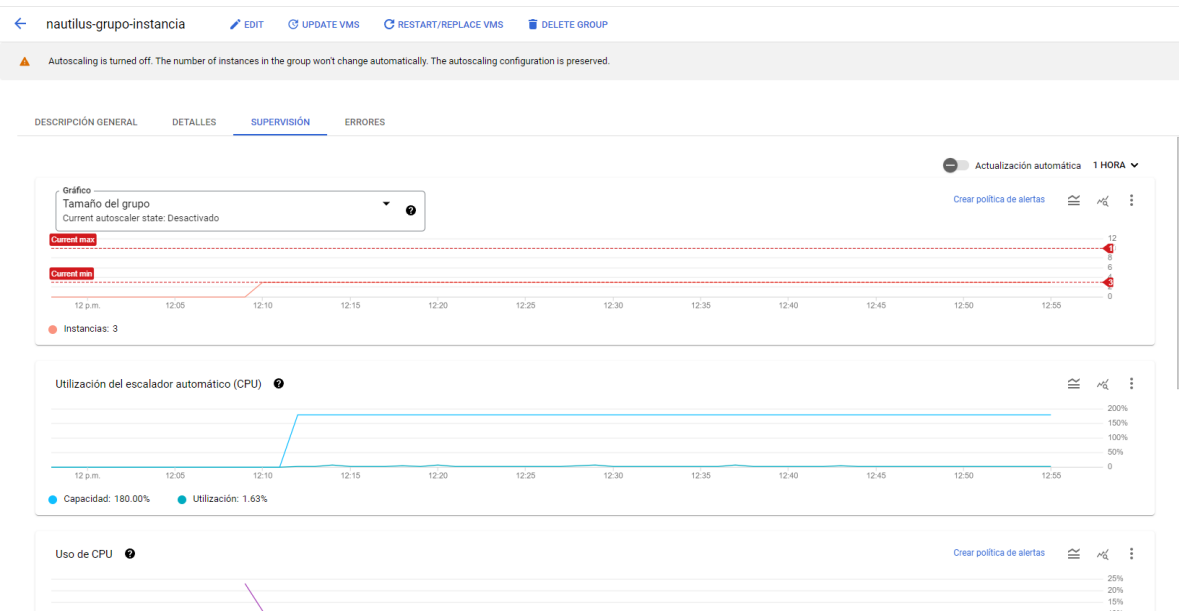
Para Nautilus Corp, es importante contar con alta disponibilidad, redundancia de sus datos y brindar protección a sus datos, es por eso que se considera la opción de RAID 6 aunque tenga un tamaño de escritura y lectura menor y una velocidad de escritura más baja, ya que la prioridad es brindar disponibilidad. 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 de costos a largo plazo y reducir el riesgo de inactividad de estos servidores.

1.5 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.

1.6 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 equitativa 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:

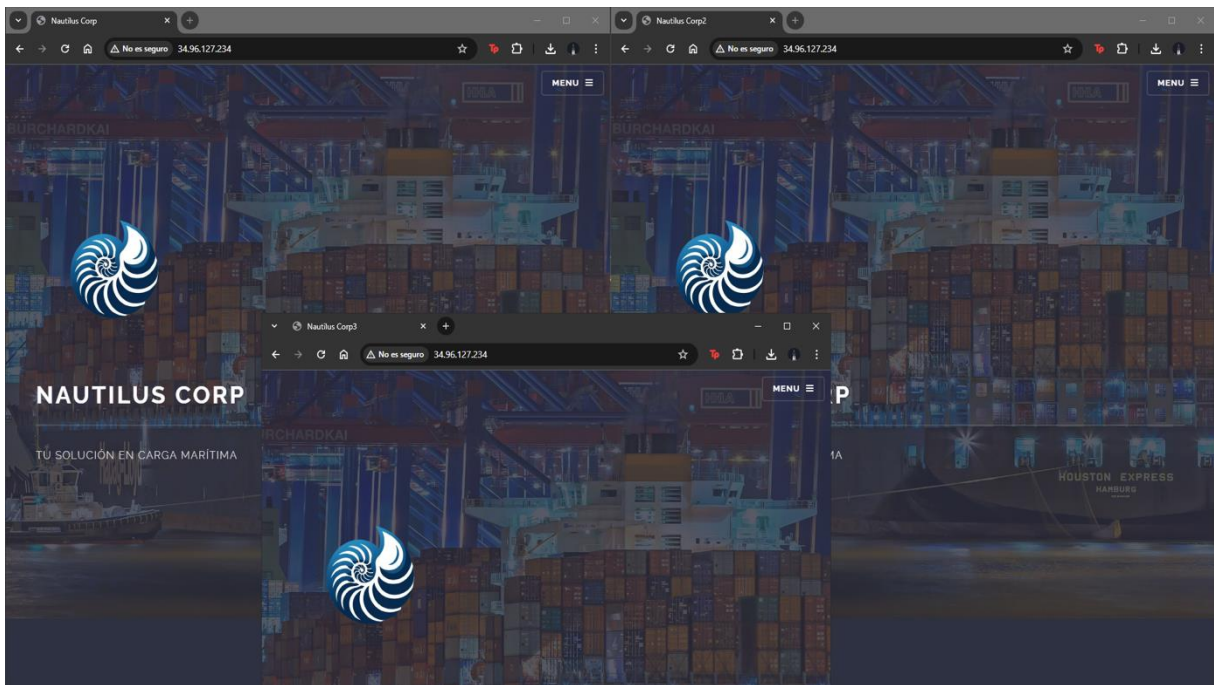


Imagen 1-G: (Propia, 2024) Se muestran las capturas del funcionamiento de la solución para los servidores web.

← nautilus-instance-group

EDITUPDATE VMSESTOP/REPLACE VMSEDELETE GROUP

Autoscaling is turned off. The number of instances in the group won't change automatically. The autoscaling configuration is preserved.

DESCRIPCIÓN GENERALDETALLESSUPERVISIÓNERRORES

Instancias por condición3 instancias3

Instancia por estadoNo configuradaReparación automática desactivadaConfigurar

Ajuste de escala automáticoDesactivadoConfigurar

StatusReady

Creation TimeMay 18, 2024, 11:01:53 p. m. UTC-06:00

Description

Target running size0

Templatenautilus-plantilla-instancias (Regional)

Locationus-central1-c

Solicitudes de cambio de tamañoNone

Instancias de VM

SUSPENDERDETENERINICIAR/REANUDARQUITAR DEL GRUPOBORRAR

FiltroIngresar el nombre o el valor de la propiedad

Estado	Nombre ↑	Fecha/hora de creación	Plantilla	Configuración por instancia	IP interna	IP externa	Estado de la verificación de estado	Conectar
<input type="checkbox"/>	<input type="radio"/> nautilus-instance-group-52v5	May 18, 2024, 11:02:06 p. m. UTC-06:00	nautilus-plantilla-instancias (Regional)		10.128.0.5 (nic0)	Ninguna		SSH
<input type="checkbox"/>	<input type="radio"/> nautilus-instance-group-59ib	May 18, 2024, 11:02:06 p. m. UTC-06:00	nautilus-plantilla-instancias (Regional)		10.128.0.6 (nic0)	Ninguna		SSH
<input type="checkbox"/>	<input type="radio"/> nautilus-instance-group-xc4b	May 18, 2024, 11:02:06 p. m. UTC-06:00	nautilus-plantilla-instancias (Regional)		10.128.0.7 (nic0)	Ninguna		SSH

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

2. SERVIDORES DE BASE DE DATOS

Nautilus Corp. También decidió migrar su infraestructura de base de datos a la nube, buscando mejorar la disponibilidad, escalabilidad y seguridad de sus datos. Los servidores de bases de datos son fundamentales para gestionar eficientemente grandes volúmenes de información y soportar las aplicaciones críticas de la empresa. A continuación, se detallan los requerimientos, la configuración y las recomendaciones para la implementación de los servidores de bases de datos para la empresa.

2.1 CONFIGURACIÓN APLICADA A LOS SERVIDORES A NIVEL DE TIPO DE MÁQUINAS, DIMENSIONAMIENTO DE DISCOS DUROS, MEMORIA RAM Y PROCESADORES:

Para montar los servidores de bases de datos en GCP, se recomienda utilizar instancias de bases de datos gestionadas que ofrezcan alta disponibilidad, escalabilidad y seguridad. El tipo de máquina recomendado es de la serie N1 para obtener un buen equilibrio entre costo y rendimiento, dependiendo de la carga de trabajo, se pueden elegir configuraciones como n1-standard-4 la cual cuenta con 4 CPU virtuales y 15 GB de RAM, especiales para bases de datos de tamaño medio. Estas configuraciones proporcionan un buen equilibrio entre costo y rendimiento, asegurando que las bases de datos de Nautilus Corp. puedan manejar eficientemente las cargas de trabajo actuales y futuras. Se presenta la solución con un costo mensual estimado de USD 110.45 el cual equivale a USD 0.15 por hora.

Nombre *
servidores-de-almacenamiento

ADMINISTRAR ETIQUETAS DE INSTANCIA Y ETIQUETAS DE RECURSO

Región *
us-west4 (Las Vegas)

Zona *
us-west4-b

La región es permanente

La zona es permanente

Configuración de la máquina

De uso general

Optimizado para procesamiento

Con optimización de memoria

Optimizada para almacenamiento

NUEVO

GPU

Tipos de máquinas para cargas de trabajo comunes, optimizados en función del costo y la flexibilidad

Series	Descripción	vCPUs	Memory	Plataforma
<input type="radio"/> N4	Flexible y con optimización de costos	2 - 80	4 - 640 GB	Emerald Rapids de Intel
<input type="radio"/> C3	Rendimiento alto y coherente	4 - 176	8 - 1,408 GB	Intel Sapphire Rapids
<input type="radio"/> C3D	Rendimiento alto y constante	4 - 360	8 - 2,880 GB	AMD Genoa
<input type="radio"/> E2	Procesamiento diario de bajo costo	0.25 - 32	1 - 128 GB	Según la disponibilidad
<input type="radio"/> N2	Precio y rendimiento equilibrados	2 - 128	2 - 864 GB	Intel Cascade y Ice Lake
<input type="radio"/> N2D	Precio y rendimiento equilibrados	2 - 224	2 - 896 GB	AMD EPYC
<input type="radio"/> T2A	Cargas de trabajo de escalamiento horizontal	1 - 48	4 - 192 GB	Altra de Ampere basada e
<input type="radio"/> T2D	Cargas de trabajo de escalamiento horizontal	1 - 60	4 - 240 GB	AMD EPYC Milan
<input checked="" type="radio"/> N1	Precio y rendimiento equilibrados	0.25 - 96	0.6 - 624 GB	Intel Skylake

Estimación mensual

USD110.45

Equivale a alrededor de USD0.15 por hora

Paga por lo que usas, con facturación por segundo y sin pagos por adelantado

Elemento	Estimación mensual
4 vCPU + 15 GB memory	USD156.21
Disco persistente balanceado de 10 GB	USD1.10
Descuento por uso	-USD46.86
Total	USD110.45

[Precios de Compute Engine](#)

[LESS](#)

Imagen 2-A: (Propia, 2024) Se muestra la configuración aplicada a los servidores de tipo de máquina, memoria RAM y procesadores.

Cuentas de servicio

Cuenta de servicio

Compute Engine default service account

Requiere que se configure el rol de usuario de cuenta de servicio (roles/iam.serviceAccountUser) para los usuarios que desean acceder a las VMs con esta cuenta de servicio. [Más información](#)

Permisos de acceso

☒ Permitir el acceso predeterminado

☐ Permitir el acceso total a todas las APIs de Cloud

☐ Configurar acceso para cada API

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

Observabilidad: Agente de operaciones

Supervisa tu sistema mediante la recopilación de registros y métricas clave.

☐ Instala el Agente de operaciones para Supervisión y Registros

Opciones avanzadas

Herramientas de redes

Interfaces de red y nombre de host

Discos

Discos adicionales

Nuevo disco

disk-1, en blanco, 10 GB

+ AGREGAR DISCO NUEVO

+ VINCULAR UN DISCO EXISTENTE

+ AGREGAR UN SSD LOCAL

Estimación mensual

USD111.55

Equivale a alrededor de USD0.15 por hora

Paga por lo que usas, con facturación por segundo y sin pagos por adelantado

Elemento	Estimación mensual
4 vCPU + 15 GB memory	USD156.21
Disco persistente balanceado de 10 GB	USD1.10
Disco persistente balanceado de 10 GB	USD1.10
Descuento por uso	-USD46.86
Total	USD111.55

[Precios de Compute Engine](#)

[^ LESS](#)

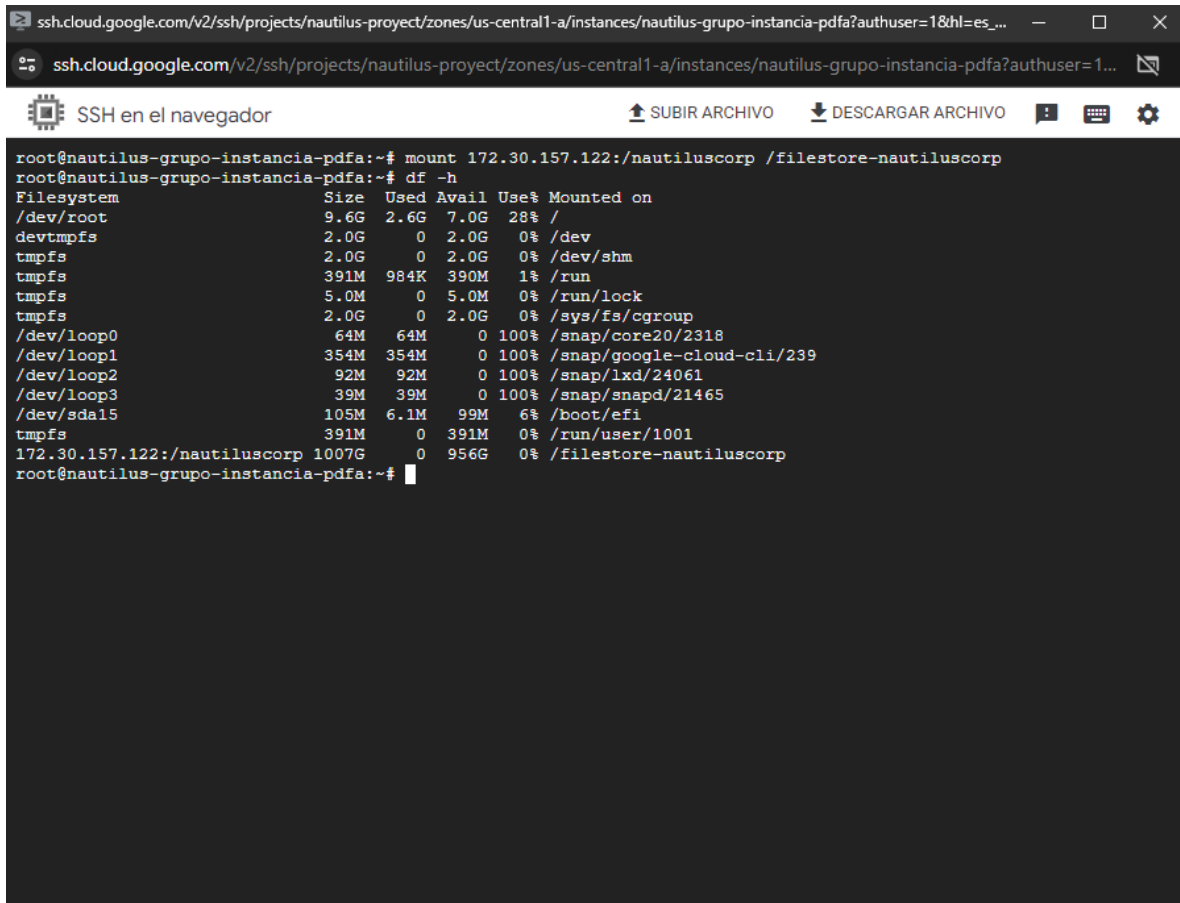
Imágen 2-B: (Propia, 2024) Se muestra la configuración aplicada a los servidores a nivel de Discos Duros.

Se aplica una configuración a nivel de discos duros por el uso del filestore. Ya que utilizar Firestore en servidores de bases de datos proporciona numerosos beneficios en términos de rendimiento, disponibilidad, escalabilidad y gestión simplificada. La configuración a nivel de discos duros mediante el uso de Filestore permite a la compañía asegurar que sus bases de datos funcionen de manera eficiente y confiable, adaptándose fácilmente a las demandas de crecimiento y cambiantes necesidades de almacenamiento.

2.2 CONFIGURACIÓN DEL FILESTORE Y SCRIPT DE ASOCIACIÓN EN SISTEMA OPERATIVO.

```
1 sudo apt update && sudo apt upgrade
2 sudo apt install mysql-server
3 mkdir /filestore-nautiluscorp
4 mount 10.40.69.82:/nautiluscorp /filestore-nautiluscorp
5 # Montar Filestore
6 sudo mkdir -p /filestore-datos
7 sudo mount -t nfs 10.40.69.82:/almacenaje /filestore-datos
8
9 # Hacer que el montaje sea persistente
10 sudo nano /etc/fstab
11 # Añadir la línea:
12 # 10.40.69.82:/almacenaje /filestore-datos nfs defaults 0 0
13 sudo mount -a
14
15 # Ajustar permisos
16 sudo chown -R mysql:mysql /filestore-datos
17 sudo chmod -R 755 /filestore-datos
18
19 # Copiar datos existentes (si los hay)
20 sudo rsync -av /var/lib/mysql/ /filestore-datos/
21 sudo chown -R mysql:mysql /filestore-datos
22
23 # Editar configuración de MySQL
24 sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf
25 # Cambiar datadir a /filestore-datos
26
27 # Configuración de AppArmor
28 sudo nano /etc/apparmor.d/tunables/alias
29 # Añadir línea: alias /var/lib/mysql/ -> /filestore-datos/,
30 sudo systemctl restart apparmor
31
32 # Reiniciar MySQL
33 sudo systemctl restart mysql
34
35 # Verificar estado
36 sudo systemctl status mysql
37
38 # Revisar registros si hay errores
39 sudo journalctl -xe
40 sudo tail -n 100 /var/log/mysql/error.log
```

Imagen 2-C: (Propia, 2024) Configuración del script.



```
ssh.cloud.google.com/v2/ssh/projects/nautilus-proyect/zones/us-central1-a/instances/nautilus-grupo-instancia-pdfa?authuser=1&hl=es_...
ssh.cloud.google.com/v2/ssh/projects/nautilus-proyect/zones/us-central1-a/instances/nautilus-grupo-instancia-pdfa?authuser=1...
SSH en el navegador SUBIR ARCHIVO DESCARGAR ARCHIVO
root@nautilus-grupo-instancia-pdfa:~# mount 172.30.157.122:/nautiluscorp /filestore-nautiluscorp
root@nautilus-grupo-instancia-pdfa:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root        9.6G  2.6G  7.0G  28% /
devtmpfs         2.0G   0  2.0G   0% /dev
tmpfs            2.0G   0  2.0G   0% /dev/shm
tmpfs            391M  984K  390M   1% /run
tmpfs            5.0M   0  5.0M   0% /run/lock
tmpfs            2.0G   0  2.0G   0% /sys/fs/cgroup
/dev/loop0        64M   64M   0 100% /snap/core20/2318
/dev/loop1       354M  354M   0 100% /snap/google-cloud-cli/239
/dev/loop2        92M   92M   0 100% /snap/lxd/24061
/dev/loop3        39M   39M   0 100% /snap/snapd/21465
/dev/sda15       105M   6.1M   99M   6% /boot/efi
tmpfs             391M   0  391M   0% /run/user/1001
172.30.157.122:/nautiluscorp 1007G   0  956G   0% /filestore-nautiluscorp
root@nautilus-grupo-instancia-pdfa:~#
```

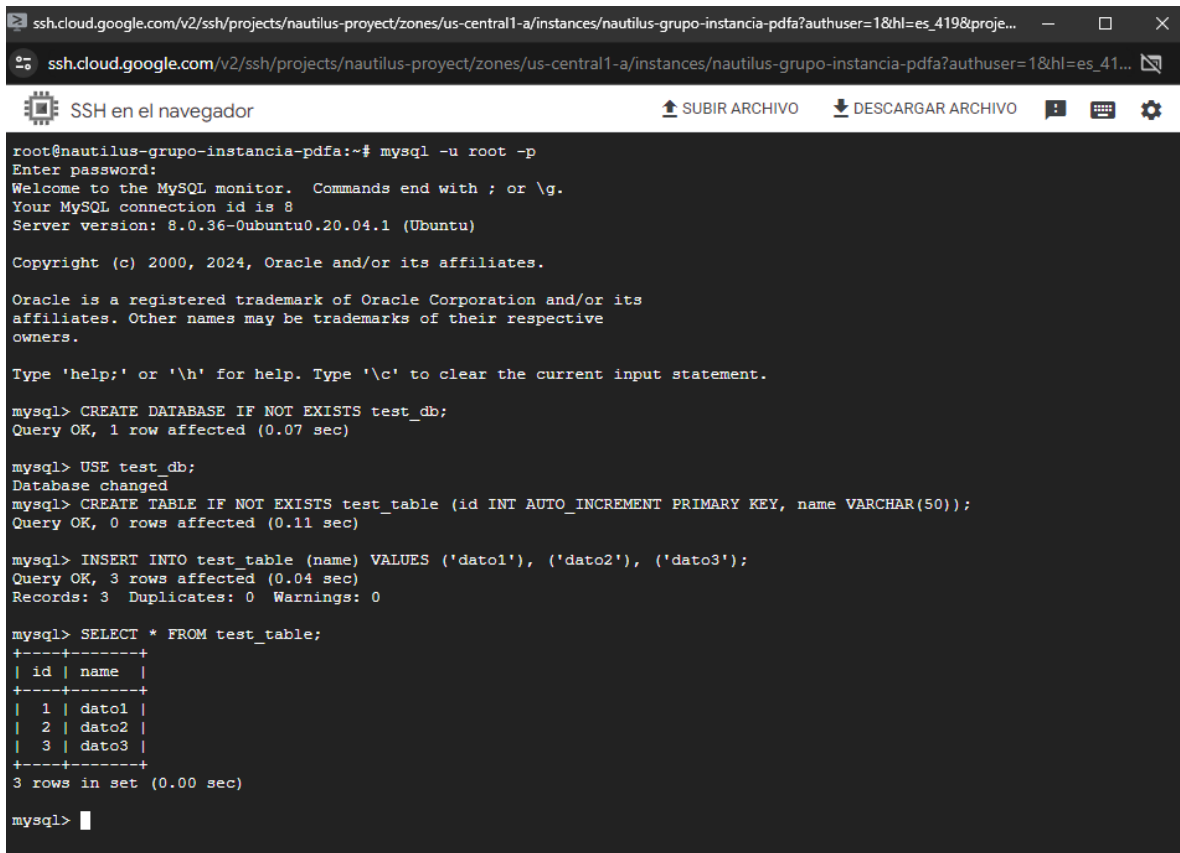
Imágen 2-D: (Propia, 2024) Configuración del Filestore.

2.3 JUSTIFICACIÓN DE LA DECISIÓN EN CUANTO AL USO DE LA BASE DE DATOS A DESPLEGAR EN EL FILESTORE.

Para la migración de bases de datos de Nautilus Corp., se ha optado por MySQL Server junto con Filestore en Google Cloud Platform debido a su balance entre costo, rendimiento y facilidad de gestión. Este servidor es de código abierto y sin costos de licencia lo que lo hace ideal para entornos de desarrollo y de pruebas. En rendimiento, MYSQL es conocido por manejar grandes cantidades de datos y alta concurrencia de uso, también permite una rápida instalación y gestión eficiente. Este ofrece escalabilidad vertical y horizontal soportando replicación para alta disponibilidad y distribución de la carga de trabajo.

Filestore complementa a MySQL proporcionando almacenamiento en red de alta disponibilidad y rendimiento, con baja latencia y alta velocidad de I/O, mejorando los tiempos de respuesta de las consultas. La replicación automática de Filestore asegura la disponibilidad continua de los datos y una recuperación rápida en caso de fallos. Además, Filestore permite escalar el almacenamiento sin interrupciones y gestiona el mantenimiento y las actualizaciones, reduciendo la carga operativa. Esta combinación ofrece una solución escalable y económicamente viable para Nautilus Corp. Asegurando un rendimiento óptimo y de alta disponibilidad con una gestión simplificada en sus bases de datos.

2.4 CAPTURAS DE IMPLEMENTACIÓN Y FUNCIONAMIENTO DE LA SOLUCIÓN.



The screenshot shows a terminal window with the title "SSH en el navegador". The user is logged into a server named "nautilus-grupo-instancia-pdf". The terminal shows the following commands and output:

```
root@nautilus-grupo-instancia-pdf:~# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.36-0ubuntu0.20.04.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE IF NOT EXISTS test_db;
Query OK, 1 row affected (0.07 sec)

mysql> USE test_db;
Database changed
mysql> CREATE TABLE IF NOT EXISTS test_table (id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(50));
Query OK, 0 rows affected (0.11 sec)

mysql> INSERT INTO test_table (name) VALUES ('dato1'), ('dato2'), ('dato3');
Query OK, 3 rows affected (0.04 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM test_table;
+----+-----+
| id | name |
+----+-----+
| 1  | dato1 |
| 2  | dato2 |
| 3  | dato3 |
+----+-----+
3 rows in set (0.00 sec)

mysql>
```

Imágen 2-E: (Propia, 2024) Captura de funcionamiento de la base de datos.



The screenshot shows a terminal window with the title "SSH en el navegador". The user is logged into a server named "nautilus-grupo-instancia-pdf". The terminal shows the following command and output:

```
root@nautilus-grupo-instancia-pdf:~# ls /filestore-nautiluscorp
'#ib_16384_0.dblwr'  binlog.000003  client-key.pem  mysql.ibd  sys
'#ib_16384_1.dblwr'  binlog.000004  debian-5.7.flag  nautilus-grupo-instancia-pdf.pid  test_db
'#innodb_redo'      binlog.000005  ib_buffer_pool  performance_schema  undo_001
'#innodb_temp'      binlog.index   ibdata1        private_key.pem      undo_002
auto.cnf            ca-key.pem     ibtmp1         public_key.pem
binlog.000001       ca.pem         lost+found     server-cert.pem
binlog.000002       client-cert.pem mysql           server-key.pem
```

Imágen 2-F: (Propia, 2024) Captura de Implementación de la base de datos.

3. 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.

3.1 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:

```

15 index.js > ...
1  const functions = require('@google-cloud/functions-framework');
2  const fs = require('fs');
3  const path = require('path');
4
5  // Función para servir el portal informativo
6  functions.http('helloHttp', (req, res) => {
7    const url = req.url;
8
9    // Si la solicitud es para el archivo HTML
10   if (url === '/') {
11     // Leer el archivo HTML del portal informativo
12     fs.readFile(__dirname + '/projects/frontend/html-serverless/redes.html', (err, data) => {
13       if (err) {
14         res.writeHead(500, {'Content-Type': 'text/plain'});
15         res.end('Error interno del servidor');
16         return;
17       }
18       res.writeHead(200, {'Content-Type': 'text/html'});
19       res.write(data);
20       res.end();
21     });
22   } else {
23     // Si la solicitud es para un archivo estático (CSS, imágenes, etc.)
24     const filePath = path.join(__dirname, '/projects/frontend/html-serverless', url);
25     fs.readFile(filePath, (err, data) => {
26       if (err) {
27         res.writeHead(404, {'Content-Type': 'text/plain'});
28         res.end('Archivo no encontrado');
29         return;
30       }
31
32       // Determinar el tipo de contenido basado en la extensión del archivo
33       let contentType = 'text/plain';
34       if (url.endsWith('.css')) {
35         contentType = 'text/css';
36       } else if (url.match(/\.(jpg|jpeg|png|gif)$/)) {
37         contentType = 'image/jpeg';
38       }
39
40       res.writeHead(200, {'Content-Type': contentType});
41       res.write(data);
42       res.end();
43     });
44   }
45 });
46

```

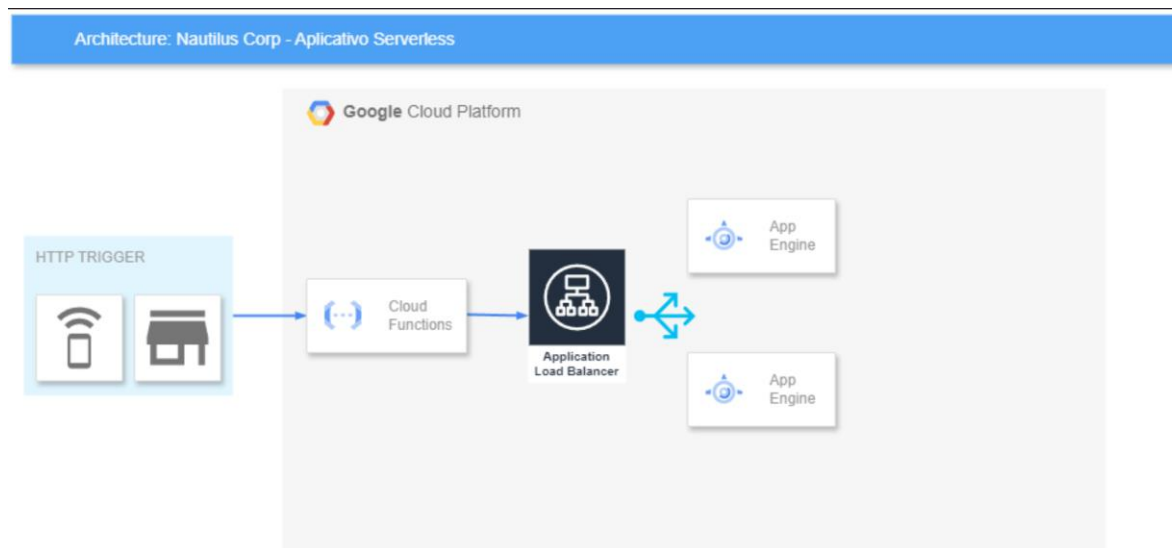
Imagen 3-A: (Propia, 2024) Se muestra el script de deploy de los servicios serverless.

3.2 ARQUITECTURA DE SERVICIOS WEB

La arquitectura de la base de datos de Nautilus Corp está ubicada en la región us-central1 y cuenta con un respaldo en otra región para garantizar la continuidad del servicio. Esta configuración incluye instancias distribuidas en diferentes zonas, con dos instancias por zona a lo largo de tres zonas, permitiendo operaciones de bases de datos altamente disponibles y el uso compartido de un único Filestore.

Los componentes clave de esta arquitectura incluyen el cliente, que representa cualquier dispositivo o software que permite a los usuarios interactuar con la plataforma en la nube; la aplicación, que abarca desde sitios web y aplicaciones web hasta juegos y software empresarial; y el motor de aplicaciones, que proporciona la infraestructura y los servicios necesarios para el funcionamiento correcto de las aplicaciones en la nube, incluyendo el sistema operativo, el servidor web, la base de datos y el middleware.

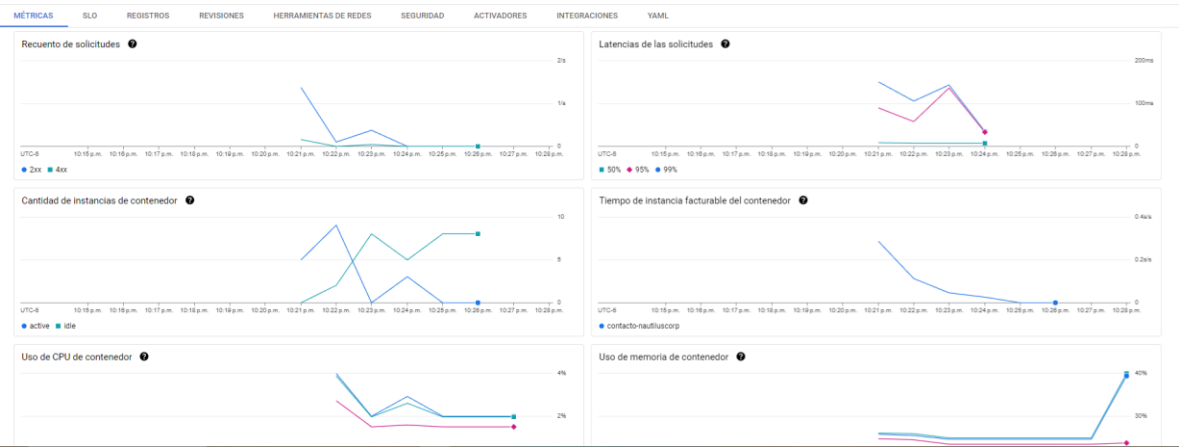
Además, se utiliza un disparador HTTP para generar eventos cuando un usuario realiza una solicitud HTTP a una aplicación específica, y funciones en la nube, que son fragmentos de código sin servidor que se ejecutan en respuesta a estos disparadores, ofreciendo una forma escalable y rentable de implementar aplicaciones en la nube. Finalmente, un equilibrador de carga de aplicaciones distribuye el tráfico entre múltiples instancias de una aplicación, asegurando alta disponibilidad y escalabilidad.



Imágen 3-B: (Propia, 2024) Se muestra la arquitectura de los servicios serverless.

3.3 ESTADÍSTICAS DE CONSUMO DE LAS INSTANCIAS BALANCEADAS

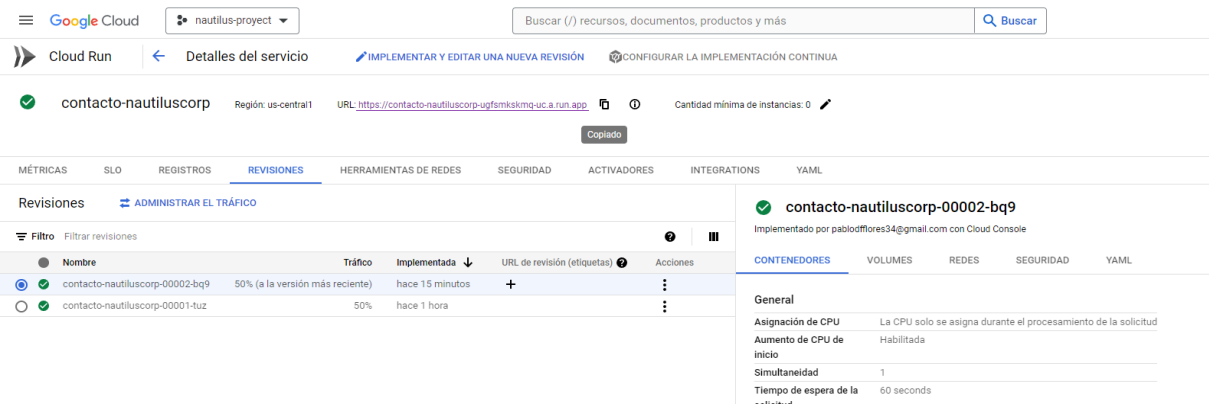
A continuación, se muestran los gráficos que proporcionan una visión integral del rendimiento y uso de los contenedores en la infraestructura de Nautilus Corp. Se muestran las fluctuaciones en el recuento de solicitudes y la cantidad de instancias de contenedor, las cuales muestran que este es un sistema dinámico que ajusta los recursos en respuesta a la carga. A continuación los resultados:



Imágen 3-C: (Propia, 2024) Se muestra la implementación de los servicios serverless.

3.4 CAPTURAS DE IMPLEMENTACIÓN Y FUNCIONAMIENTO DE LA SOLUCIÓN.

Para la solución presentada, se implementó una arquitectura serverless la cual es escalable ya que permite gestionar sus aplicaciones y servicios web. Esta cuenta con beneficios como la automatización, escalabilidad automática, reducción de costos y una gestión simplificada. A continuación, se presentan los resultados:



Imágen 3-D: (Propia, 2024) Se muestra la implementación de los servicios serverless.

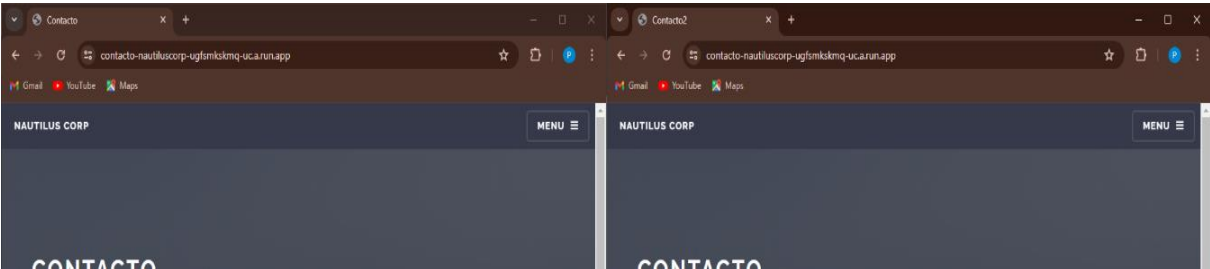


Imagen 3-E: (Propia, 2024) Se muestra el funcionamiento de los servicios serverless.

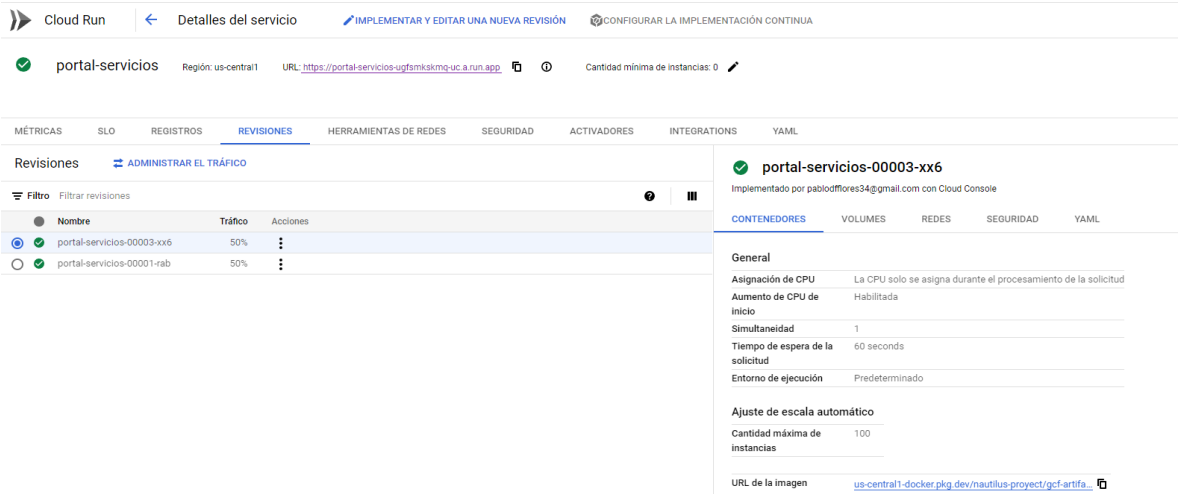


Imagen 3-F: (Propia, 2024) Se muestra la implementación de los servicios serverless.

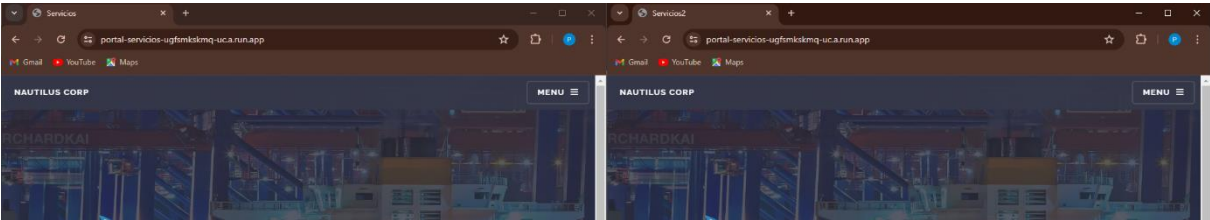
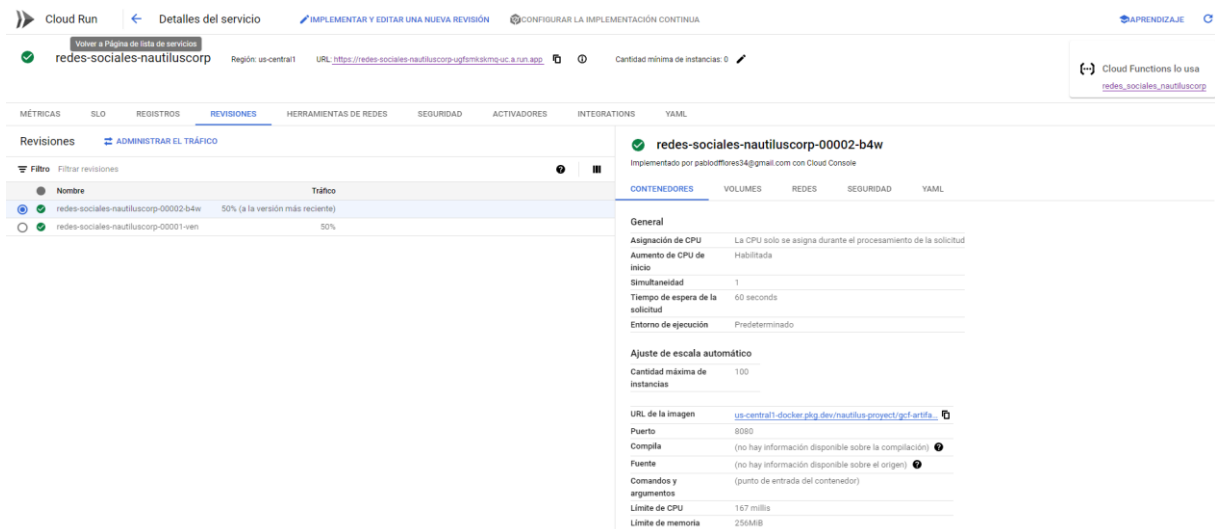
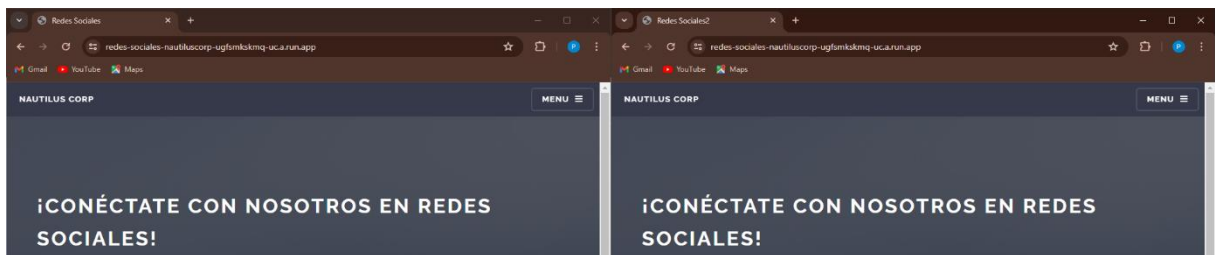


Imagen 3-G: (Propia, 2024) Se muestra el funcionamiento de los servicios serverless.



Imágen 3-H: (Propia, 2024) Se muestra la implementación de los servicios serverless.



Imágen 3-I: (Propia, 2024) Se muestra el funcionamiento de los servicios serverless.

4. SERVIDORES DE ALMACENAMIENTO – BUCKETS

4.1 TAXONOMÍA DE LOS ARBOLES DE ARCHIVOS DE AMBOS BUCKETS

De acuerdo a los archivos creados, se muestra la estructura de dos buckets en Google Cloud Storage (GCS), uno para audio y otro para video. Ambos buckets siguen una taxonomía de árbol de archivos similar, con una raíz que representa el nombre del bucket, seguido de niveles que representan categorías y subcategorías de archivos.

En el bucket de audio, el nivel 1 divide los archivos en Música, Notas de voz y Ubicaciones. El nivel 2 dentro de Música contiene subcarpetas para formatos como MP3 y WAV, mientras que Ubicaciones alberga subcarpetas como Tiempo Real y Última ubicación.

El bucket de video, por otro lado, clasifica los archivos en 4K, Full HD y HD en el nivel 1.

Esta estructura facilita la búsqueda y recuperación de archivos, ya que cada archivo se almacena en una ruta específica que refleja su categoría y subcategoría.

Cabe destacar que la taxonomía de archivos puede personalizarse según las necesidades de cada organización, y es crucial mantener una estructura consistente para una gestión eficiente de los datos. GCS ofrece herramientas para crear, eliminar y organizar carpetas, además de cargar archivos en ubicaciones específicas.

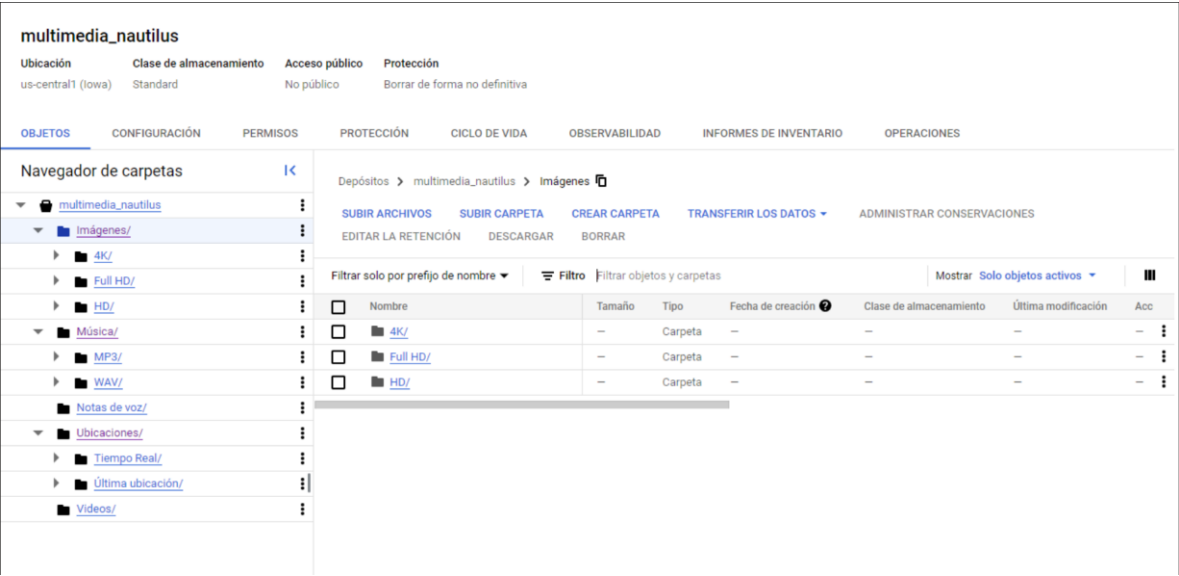


Imagen 4-A: (Propia, 2024) Se muestra la taxonomía de los árboles de archivos del bucket de multimedia.

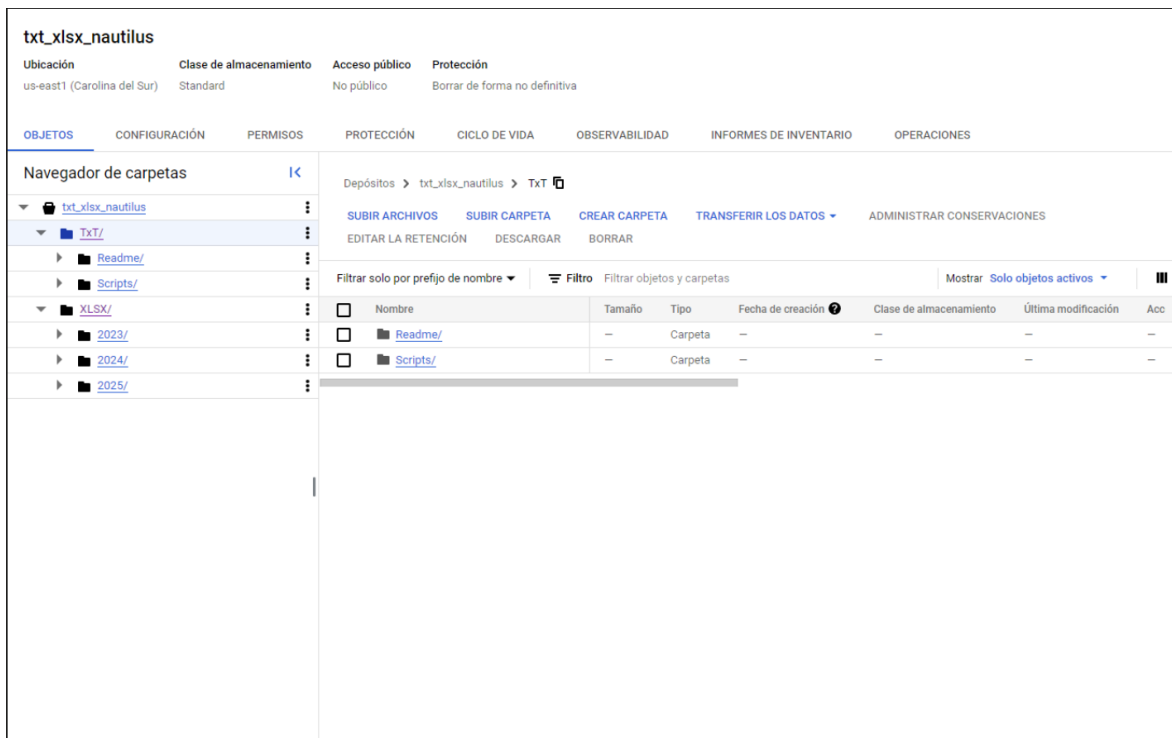


Imagen 4-B: (Propia, 2024) Se muestra la taxonomía de los árboles de archivos del bucket de txt_xlsx.

4.2 POLÍTICAS DE ACCESO – JUSTIFICACIÓN DE ROLES

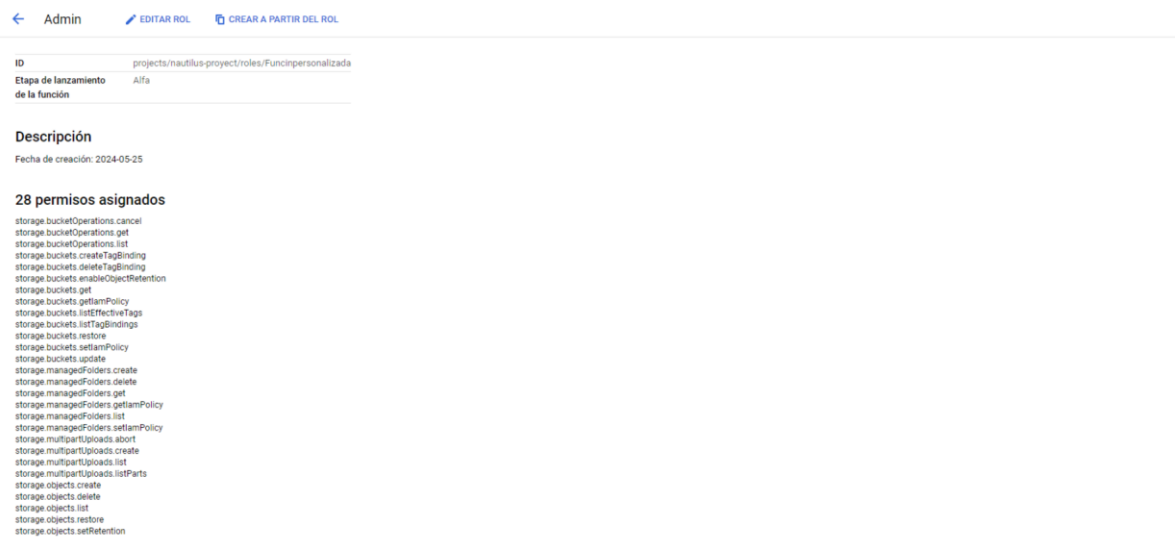
En el entorno de Google Cloud Platform (GCP), la gestión de identidades y accesos se organiza en tres roles distintos: administrador, operador y usuario. Cada uno de estos roles desempeña funciones específicas dentro del marco de la gestión de identidades IAM (Identity and Access Management), con el objetivo de garantizar la seguridad y eficiencia en el uso de los recursos de la plataforma.

El rol de administrador ocupa el nivel más alto en la jerarquía de acceso. Los administradores tienen privilegios amplios y son responsables de configurar y mantener la infraestructura de identidad y acceso en GCP. Esto incluye la creación y gestión de cuentas de usuario, la asignación de roles y permisos, y el establecimiento de políticas de seguridad para proteger los recursos de la plataforma. Los administradores son los encargados de garantizar que se apliquen las mejores prácticas de seguridad y que se cumplan los requisitos de cumplimiento normativo.

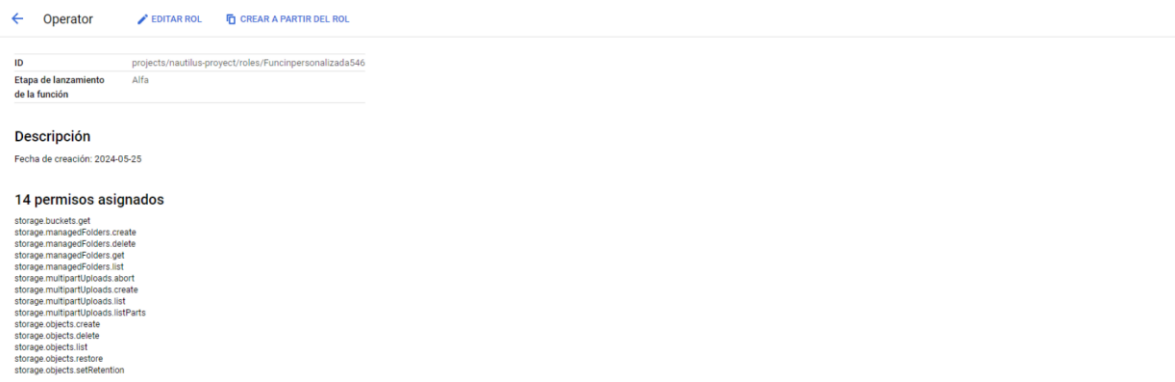
Los operadores, por su parte, ocupan un nivel intermedio en la jerarquía de acceso. Su función principal es gestionar las identidades de los usuarios en GCP, lo que implica la creación, modificación y eliminación de cuentas de usuario, así como la asignación de roles y permisos según las necesidades del negocio. Los operadores

actúan como intermediarios entre los administradores y los usuarios finales, asegurando que se apliquen las políticas de seguridad de manera adecuada y que se respeten los principios de least privilege access (LPA) para minimizar los riesgos de seguridad.

Finalmente, los usuarios representan el nivel más bajo en la jerarquía de acceso. Tienen derechos limitados y solo pueden acceder a los recursos autorizados para ellos en GCP. Los usuarios pueden realizar tareas específicas dentro de la plataforma, como utilizar aplicaciones o acceder a datos, según los permisos y roles asignados por los administradores y operadores. El principio de least privilege access (LPA) se aplica aquí, lo que significa que los usuarios solo tienen acceso a los recursos y funciones necesarios para llevar a cabo sus tareas, reduciendo así el riesgo de exposición de datos sensibles o mal uso de los recursos en GCP.



Imágen 4-C: (Propia, 2024) Se muestra los permisos asignados para un usuario de tipo Administrador.



Imágen 4-D: (Propia, 2024) Se muestra los permisos asignados para un usuario de tipo Operador.

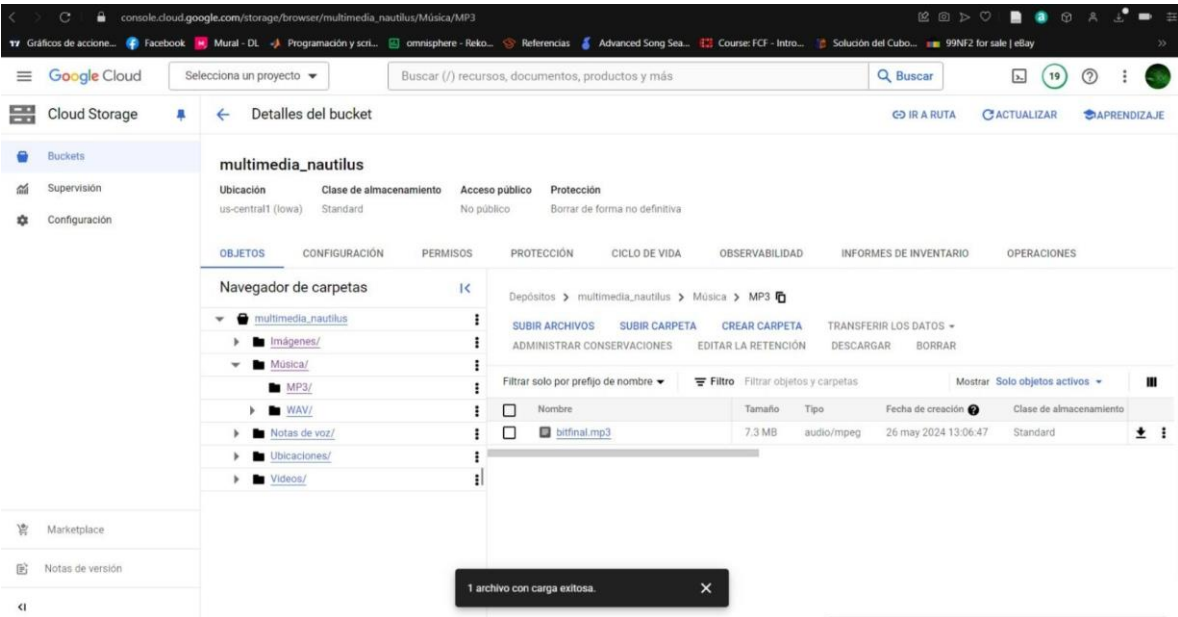


Imagen 4-E: (Propia, 2024) Se muestra la funcionalidad de un usuario tipo Operador.

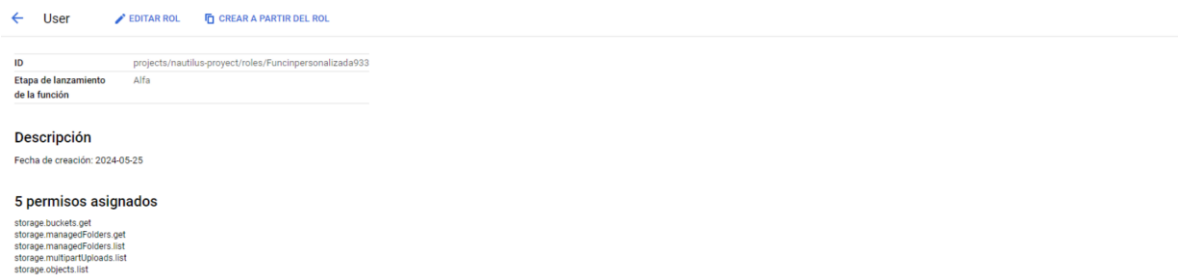
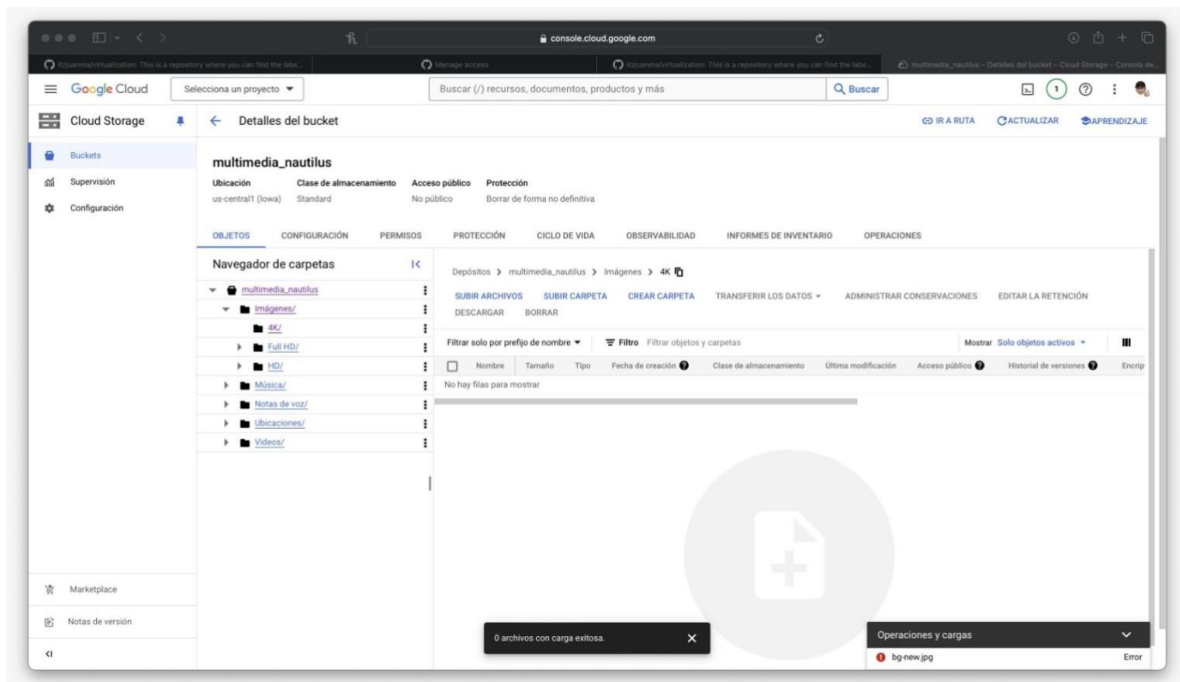
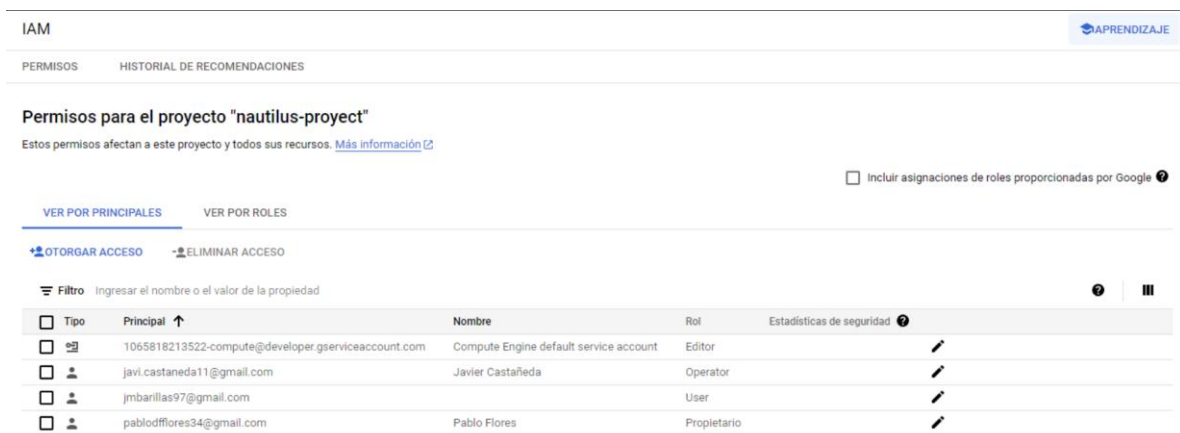


Imagen 4-F: (Propia, 2024) Se muestra los permisos asignados para un usuario de tipo User.



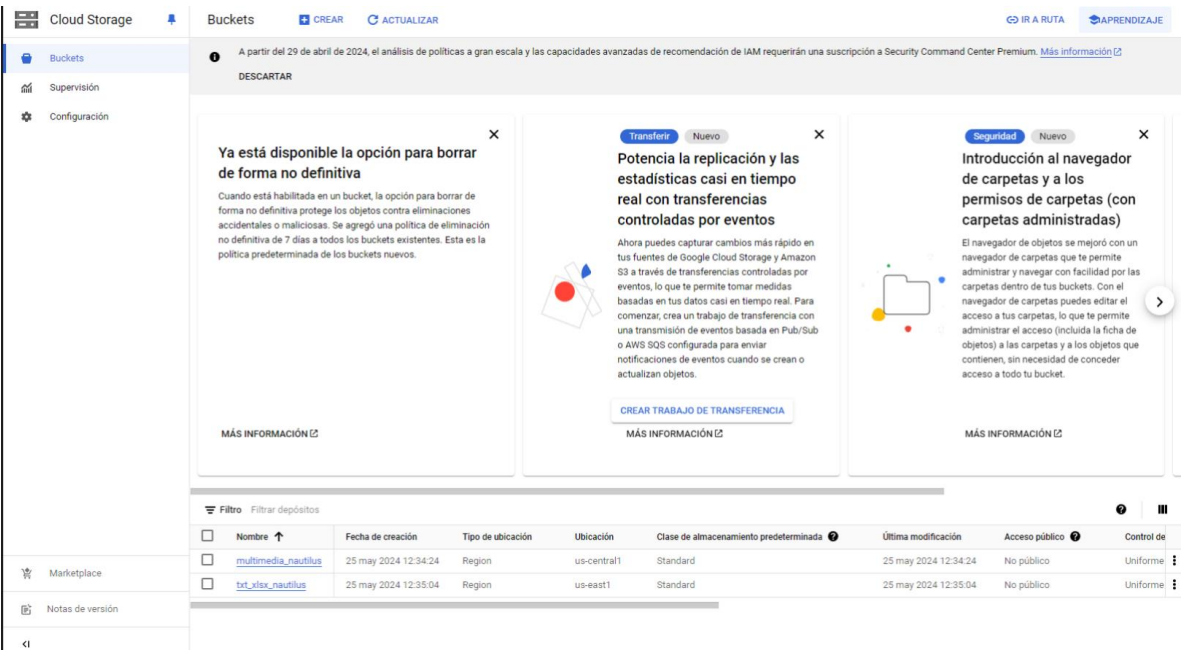
Imágen 4-G: (Propia, 2024) Se muestra la funcionalidad de un usuario tipo User.



Imágen 4-H: (Propia, 2024) Se muestran todos los usuarios con sus respectivos roles.

4.3CAPTURAS DE IMPLEMENTACIÓN Y FUNCIONAMIENTO DE LA SOLUCIÓN

En este apartado se puede observar la página de administración de buckets en Google Cloud Storage (GCS). En ella se puede observar dos buckets creados para el almacenamiento masivo de diferentes tipos de datos no estructurados.



Imágen 4-I: (Propia, 2024) Se muestran los dos buckets creados.

5. SERVICIOS DE BACKUP DE BD

La implementación del sistema de backup para Nautilus Corp. está diseñada para cumplir con estrictos requisitos de retención, seguridad y disponibilidad. Al utilizar Google Cloud Storage para el almacenamiento, Google Cloud Functions para la automatización y la configuración multirregional para la redundancia, se asegura que los datos críticos de la empresa estén protegidos contra pérdidas y accesibles en todo momento. Esta estrategia proporciona una solución robusta y escalable que soporta el crecimiento continuo y las necesidades de protección de datos de Nautilus Corp.

5.1 PLAN DE BACKUP

Cloud Scheduler	Trabajos	CREAR TRABAJO	ACTUALIZAR	FORZAR EJECUCIÓN	EDITAR	COPIAR	PAUSAR	REANUDAR	BORRAR	APRENDIZAJE
TRABAJOS DE SCHEDULER	TRABAJOS CRON DE APP ENGINE									
Filtro	Filtrar trabajos									
<input type="checkbox"/>	Nombre ↑	Estado de la última ejecución	Región	Estado	Descripción	Frecuencia	Destino	Última ejecución	Próxima publicación	Acciones
<input type="checkbox"/>	backup-executer-nautiluscorp	Aún no se ha ejecutado	us-central1	Habilitado		0 0 1 * * * (America/Godthab)	URL: https://console.cloud.google.com/storage/browser/backup-filestore-nautiluscorp			

Imágen 5-A: (Propia, 2024) Se presenta la implementación del plan de backup.

Cloud Scheduler

← Crea un trabajo

APRENDIZAJE

Volver a Página de lista de trabajos

sin costo.

LEARN MORE

Define el programa

Nombre *

backup-executer-nautiluscorp

Debe ser único en los trabajos de la misma región.

Región *

us-central1 (Iowa)

Descripción

Frecuencia *

0 0 1 * * *

Las programaciones se especifican con el formato cron de Unix. P. ej., cada minuto: * * * * *, cada 3 horas: 0 * /3 * * * *, todos los lunes a las 9:00 a.m.: 0 9 * * 1*. [Más información](#)

Minuto y hora:

A la(s) 12:00 a.m.

y día:

Igual a 1 (día del mes)

Zona horaria *

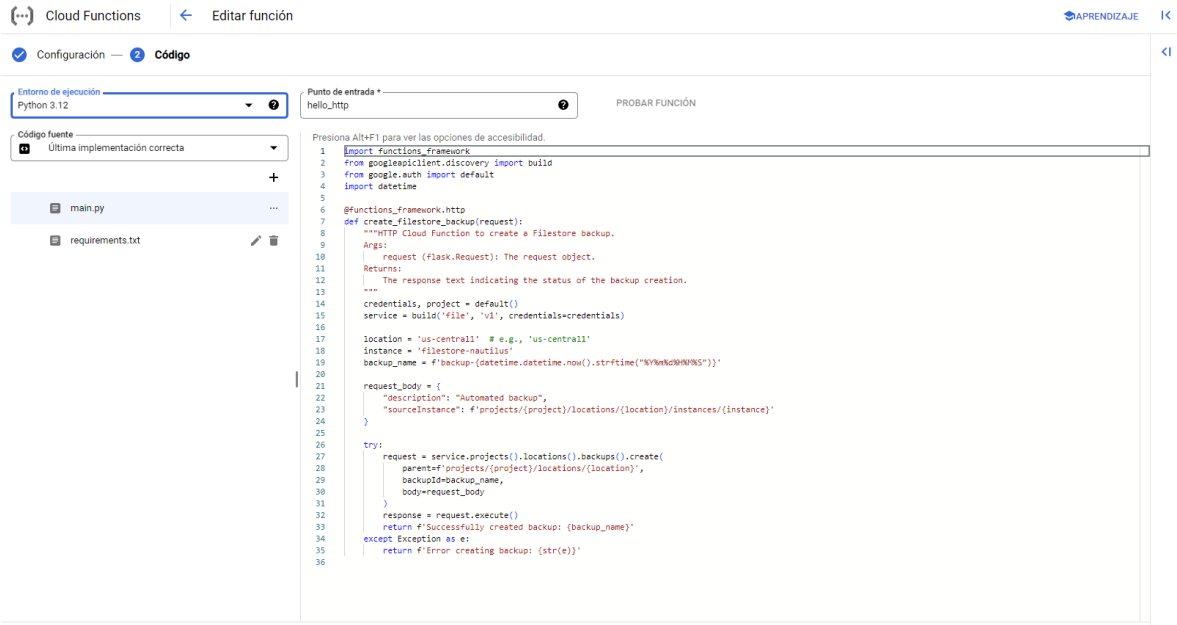
Greenland Summer Time (WGST)

Los trabajos establecidos en zonas horarias afectadas por el horario de verano pueden ejecutarse fuera de la cadencia durante el cambio a DST. Si usas una zona horaria UTC, puedes evitar el problema. [Más información](#)

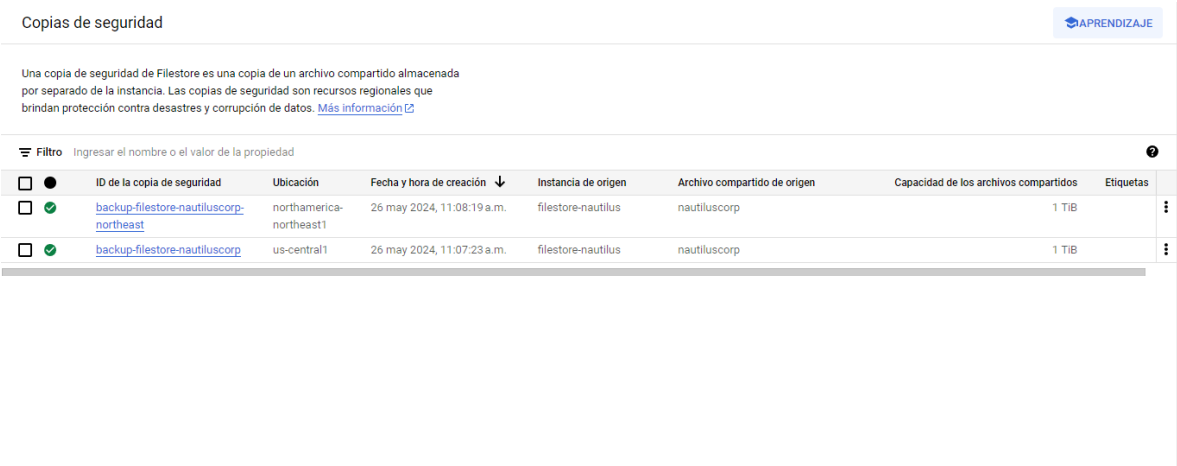
CONTINUAR

Imágen 5-B: Imágen 5 A: (Propia, 2024) Se presenta la implementación del plan de backup.

5.2 CAPTURAS DE IMPLEMENTACIÓN Y FUNCIONAMIENTO DE LA SOLUCIÓN.



Imágen 5-A: (Propia, 2024) Se presenta la implementación de la solución



Imágen 5-B: (Propia, 2024) Se presenta el funcionamiento de la solución

6. CONTENEDORES

6.1 SCRIPT DE IMPLEMENTACIÓN DEL CONTENEDOR

```
root@nautilus-grupo-instancia-pdfa:~/docker-images/docker-images/virtualization# cd ..
root@nautilus-grupo-instancia-pdfa:~/docker-images/docker-images# cd ..
root@nautilus-grupo-instancia-pdfa:~/docker-images# nano Dockerfile
root@nautilus-grupo-instancia-pdfa:~/docker-images# docker build --tag apache-centos:capache-copy .
[+] Building 3.1s (8/8) FINISHED                                docker:default
=> [internal] load build definition from Dockerfile              0.0s
=> => transferring dockerfile: 172B                             0.0s
=> [internal] load metadata for docker.io/library/centos:7      0.4s
=> [internal] load .dockerignore                                0.0s
=> => transferring context: 2B                                   0.0s
=> [1/3] FROM docker.io/library/centos:7@sha256:be65f488b7764ad3638f236b7b515b3678369a5124c47b8d32916d6487418e 0.0s
=> [internal] load build context                                1.0s
=> => transferring context: 63.66MB                             1.0s
=> CACHED [2/3] RUN yum install httpd -y                        0.0s
=> [3/3] COPY docker-images/virtualization/projects/frontend /var/www/html 1.1s
=> exporting to image                                           0.5s
=> => exporting layers                                           0.5s
=> => writing image sha256:73dc4d0d44ff4b5915f48f67baf5d5947f17546d94ebfd614e336eebce518ab0 0.0s
=> => naming to docker.io/library/apache-centos:capache-copy 0.0s
root@nautilus-grupo-instancia-pdfa:~/docker-images# docker run -d --name apache -p 80:80 apache-centos:capache-copy
61e1b3f131dddb1c923bcb82e643c934d9a162cc3f48eda3e25cb89e4cb6423f
root@nautilus-grupo-instancia-pdfa:~/docker-images# docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS
61e1b3f131dd   apache-centos:capache-copy          "/bin/sh -c 'apachec..." 4 seconds ago  Up 2 seconds  0.0.0.0:80->80/tcp
, :::80->80/tcp   apache
```

Imagen 6-A: (Propia, 2024) Se presenta el script de la implementación del contenedor.

6.2 CAPTURA DE CONFIGURACIÓN DE MÁQUINA DE IMPLEMENTACIÓN

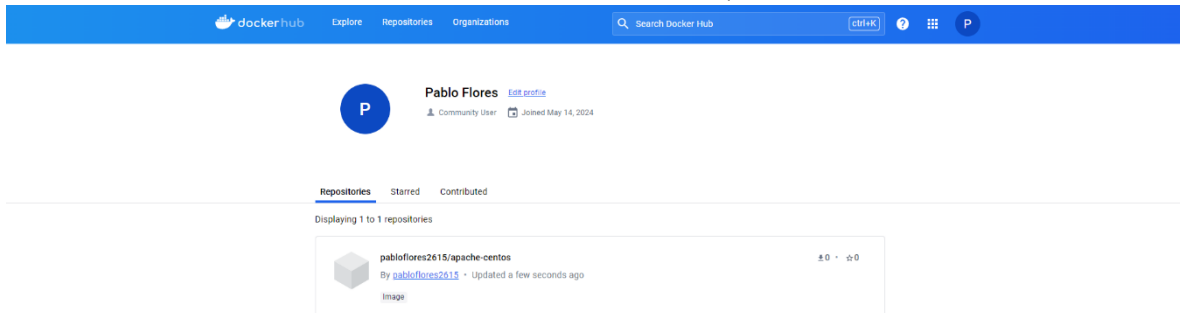
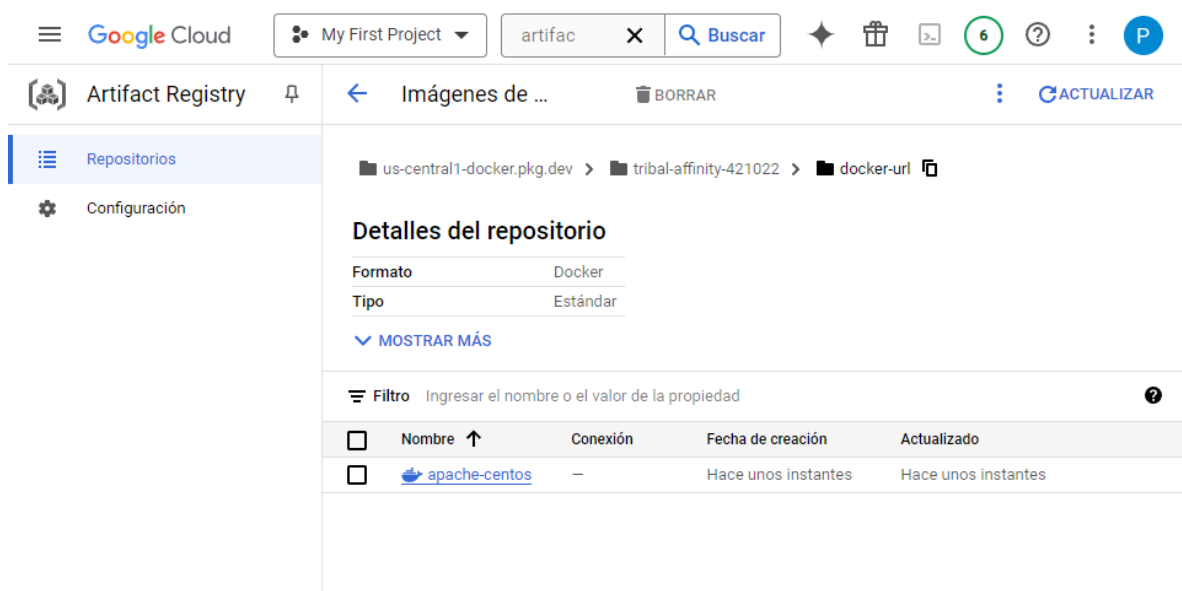


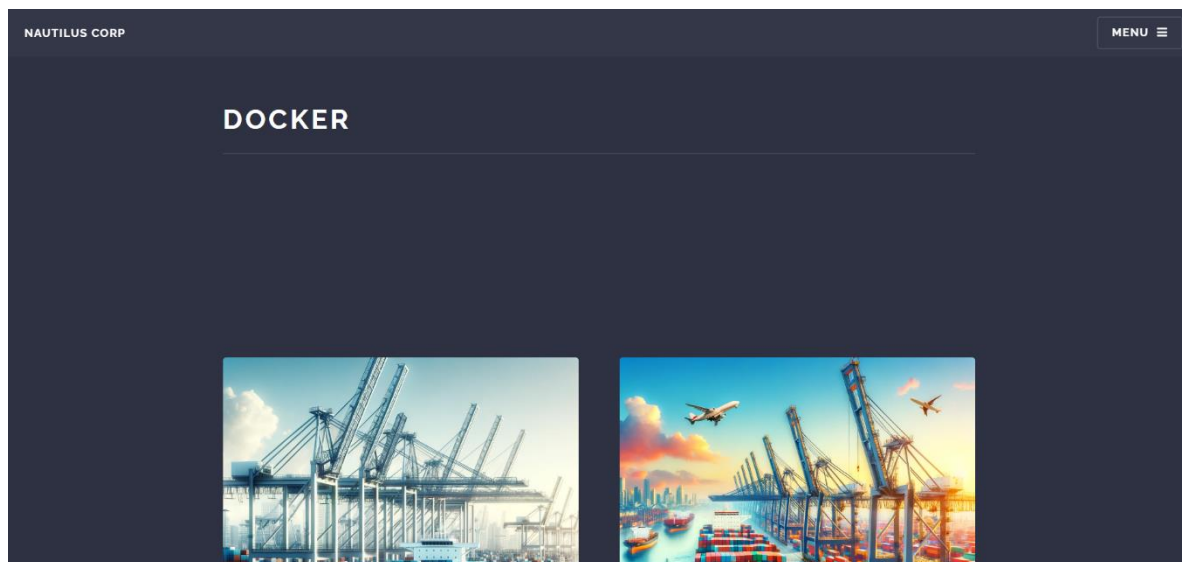
Imagen 6-B: (Propia, 2024) Se presenta el script de la configuración del contenedor.

6.3 CAPTURAS DE IMPLEMENTACIÓN EN LA NUBE.



Imágen 6-C: (Propia, 2024) Se presenta el script de la configuración del contenedor.

6.4 CAPTURAS DE IMPLEMENTACIÓN Y FUNCIONAMIENTO DE LA SOLUCIÓN



Imágen 6-D: (Propia, 2024) Se presenta el script de la implementación del contenedor.

ANEXOS

a. 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. reiniciar 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
```