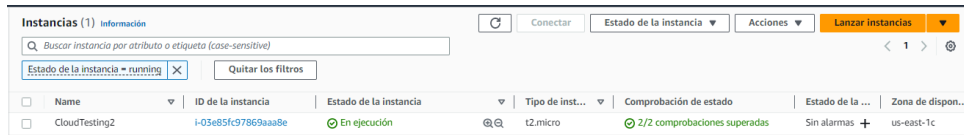


ANÁLISIS DE CAPACIDAD MIGRACIÓN DE UNA APLICACIÓN WEB A LA NUBE PÚBLICA

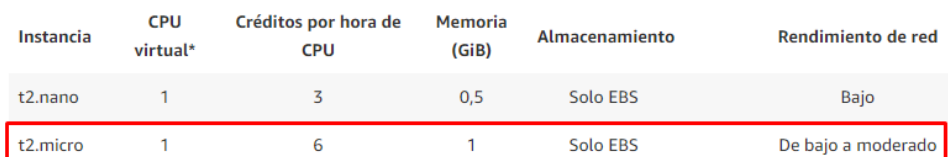
Las pruebas presentadas en este documento fueron realizadas en una instancia t2.micro de AWS.



The screenshot shows the AWS Management Console 'Instances' page. A table lists the instance 'CloudTesting2' with ID 'i-03e85fc97869aaa8e', state 'En ejecución', type 't2.micro', and zone 'us-east-1c'. The instance is running on 'us-east-1c'.

Name	ID de la instancia	Estado de la instancia	Tipo de inst...	Comprobación de estado	Estado de la ...	Zona de dispon...
CloudTesting2	i-03e85fc97869aaa8e	En ejecución	t2.micro	2/2 comprobaciones superadas	Sin alarmas	us-east-1c

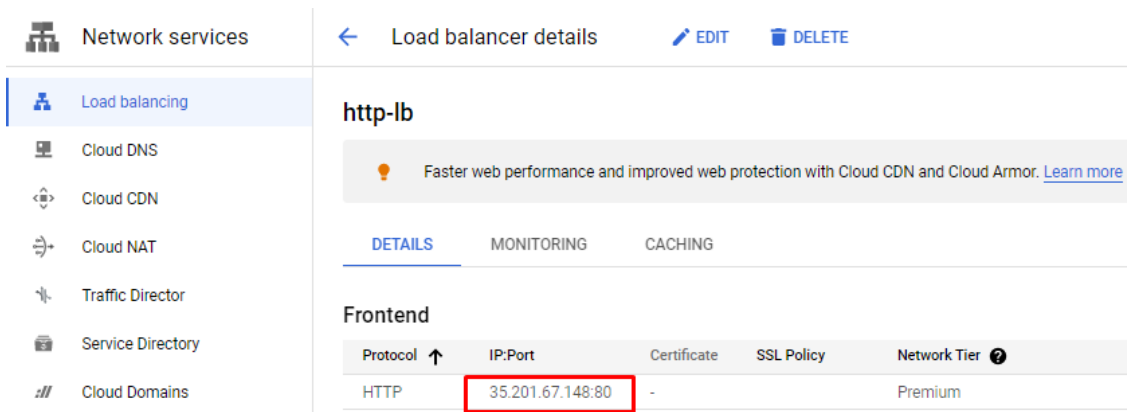
Las propiedades de este tipo de instancia se muestran en la siguiente imagen:



The screenshot shows the AWS Instance Types page. A table lists the properties for 't2.nano' and 't2.micro' instances. The 't2.micro' row is highlighted with a red border.

Instancia	CPU virtual*	Créditos por hora de CPU	Memoria (GiB)	Almacenamiento	Rendimiento de red
t2.nano	1	3	0,5	Solo EBS	Bajo
t2.micro	1	6	1	Solo EBS	De bajo a moderado

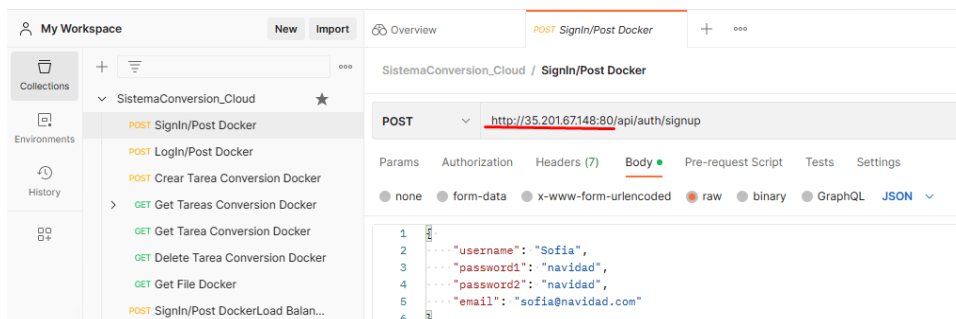
Para la realización de dichas pruebas las herramientas de Postman y Jmeter apuntaron a la IP externa del balanceador de carga creado para el grupo de instancias del web server.



The screenshot shows the AWS Network services console. The 'Load balancer details' page for 'http-lb' is displayed. The 'Frontend' tab is selected, showing a table with columns: Protocol, IP:Port, Certificate, SSL Policy, and Network Tier. The 'IP:Port' column shows '35.201.67.148:80'.

Protocol	IP:Port	Certificate	SSL Policy	Network Tier
HTTP	35.201.67.148:80	-	-	Premium

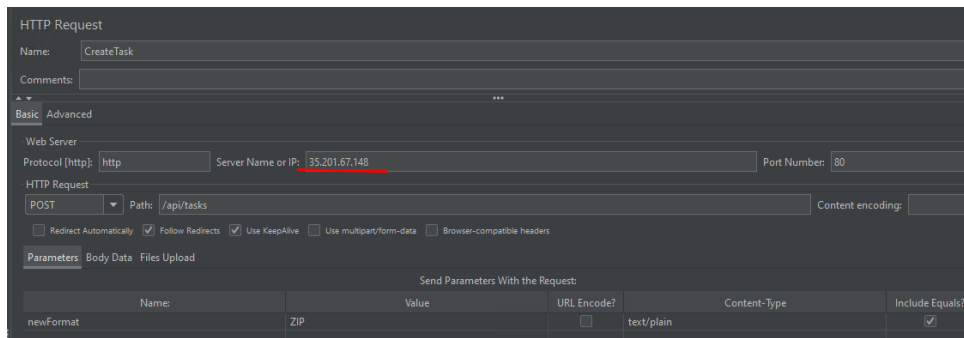
Url base Utilizada en PostMan



The screenshot shows the Postman workspace. A POST request is defined with the URL 'http://35.201.67.148:80/api/auth/signup'. The 'Body' tab is selected, showing a JSON payload with fields: 'username', 'password1', 'password2', and 'email'.

```
1 {
2   "username": "Sofia",
3   "password1": "navidad",
4   "password2": "navidad",
5   "email": "sofia@navidad.com"
6 }
```

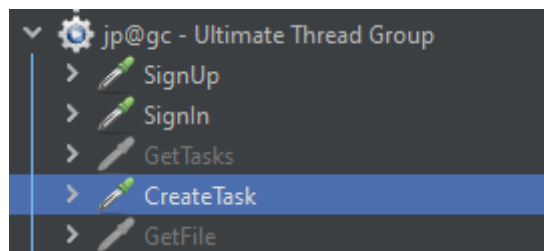
Url base Utilizada en JMeter



Se realizaron pruebas de rendimiento de acuerdo a los escenarios posteriormente descritos:

Escenario 1. Se deberá definir un escenario donde se pueda probar cuál es la máxima cantidad de requests HTTP por minuto que soporta la aplicación web con usuarios. Para hacer pruebas de estrés se debe utilizar la herramienta Apache Bench (ab) o JMeter. Las pruebas de estrés deberán realizarse desde otros equipos diferentes a los utilizados para ejecutar el servidor web y el servidor de base de datos. El escenario y los resultados de las pruebas de estrés deberán ser documentados con gráficas que ilustran cómo se comporta el sistema a medida que el número de clientes accediendo a la aplicación se incrementa hasta llegar al punto de degradar completamente el rendimiento de esta.

Para realizar esta prueba, se configuraron en JMeter las peticiones correspondientes al proceso de SignUp, SignIn, y Create Task.



Las peticiones de SignUp y SignIn eran necesarias en el grupo de peticiones ya que a partir de la información de respuesta de dichas peticiones se obtiene el requerido actualmente por la aplicación para poder realizar la petición de Create Task.

A continuación, se presentan los datos usados en las peticiones en mención:

SignUp:

Petición HTTP

Nombre:

Comentarios:

Basic Advanced

Servidor Web

Protocolo: Nombre de Servidor o IP:

Petición HTTP

Ruta:

☐ Redirigir Automáticamente ☒ Seguir Redirecciones ☒ Utilizar KeepAlive ☐ Usar 'multipart/form-data'

Parameters Body Data Files Upload

```
1 {  
2   "username": "Sofia",  
3   "password1": "navidad",  
4   "password2": "navidad",  
5   "email": "sofia@navidad.com"  
6 }  
7
```

Sign In:

Petición HTTP

Nombre:

Comentarios:

Basic Advanced

Servidor Web

Protocolo: Nombre de Servidor o IP:

Petición HTTP

Ruta:

☐ Redirigir Automáticamente ☒ Seguir Redirecciones ☒ Utilizar KeepAlive ☐ Usar 'multipart/form-data'

Parameters Body Data Files Upload

```
1 {  
2   "username": "Sofia",  
3   "password": "navidad"  
4 }  
5
```

Create Task:

Petición HTTP

POST Ruta: /api/tasks

☐ Redirigir Automáticamente ☒ Seguir Redirecciones ☒ Utilizar KeepAlive ☐ Usar 'multipart/form-data' para HTTP POST ☐ Cabeceras compatibles con navegadores

Parameters Body Data Files Upload

Enviar Parámetros Con la Petición:

Nombre:	Valor
newFormat	ZIP

Petición HTTP

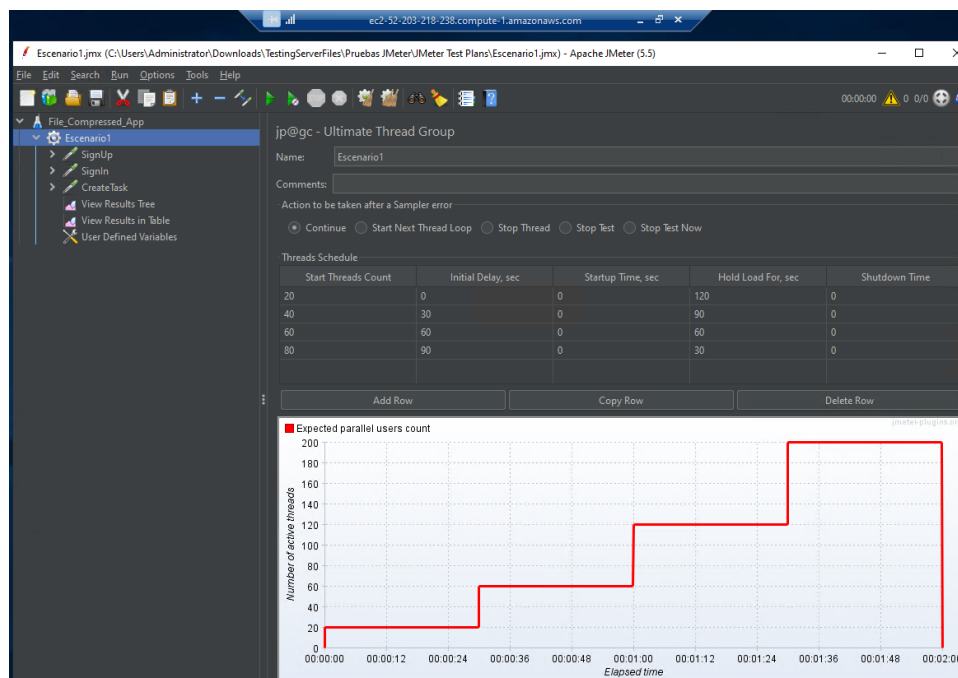
POST Ruta: /api/tasks

☐ Redirigir Automáticamente ☒ Seguir Redirecciones ☒ Utilizar KeepAlive ☐ Usar 'multipart/form-data' para HTTP POST ☐ Cabeceras compatibles con navegadores

Parameters Body Data Files Upload

Nombre de Archivo: C:\Users\Helenal\OneDrive\Documentos\Maestría_Ingeniería_Software\Desarrollo_de_Software_en_la_Nube\The shift to the Cloud computing.pdf Nombre de Parámetro: fileName

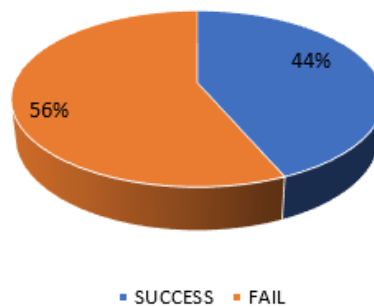
La rampa de carga utilizada en este escenario se presenta en la siguiente imagen:



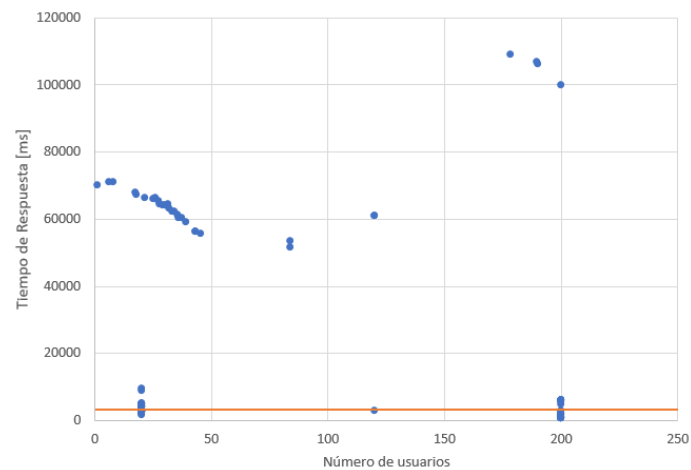
A continuación se presentan los resultados obtenidos para las pruebas realizadas en el Despliegue previo (Bucket File Storage +Autoescalning + Load Balancing) y el Despliegue Actual (Alta disponibilidad del Web Server + Servicio de Mensajes Cloud Pub/Sub + Worker con Autoscaling (Entrega Actual).

Porcentaje de Peticiones Exitosas – Despliegue Previo

Status de las Peticiones

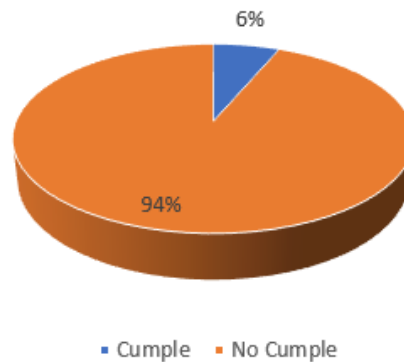


Grafica Tiempo de Respuesta Vs Usuarios – Despliegue Previo



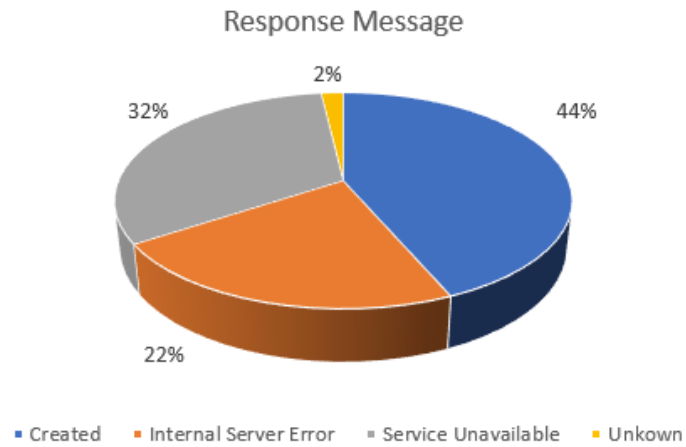
Porcentaje Cumplimiento tiempo de respuesta – Despliegue Previo¹

Porcentaje de Cumplimiento
Requerimiento Tiempo de Respuesta

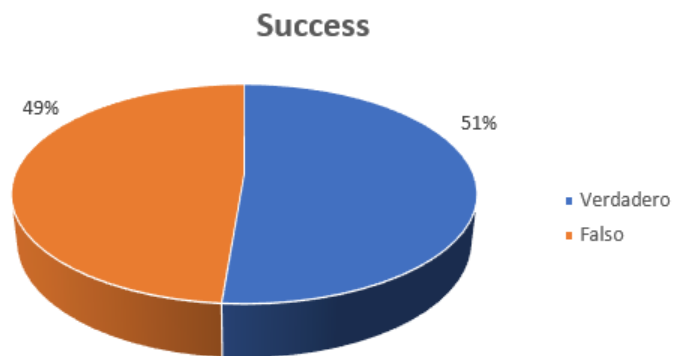


¹ Calculado considerando únicamente las peticiones exitosas

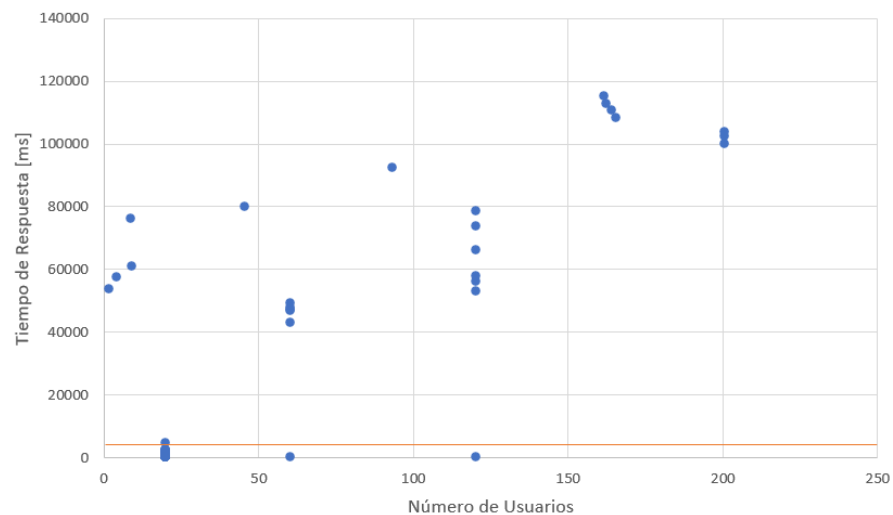
Tipos de mensajes de respuestas obtenidos durante la prueba- Despliegue Previo



Porcentaje de Peticiones Exitosas – Despliegue Actual

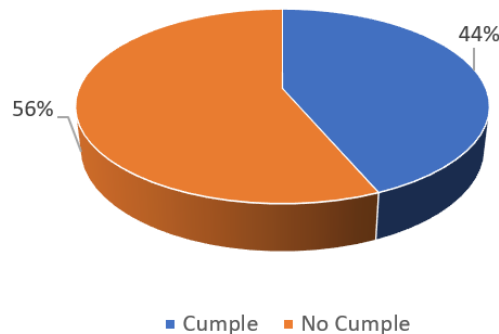


Grafica Tiempo de Respuesta Vs Usuarios – Despliegue Actual



Porcentaje Cumplimiento tiempo de respuesta – Despliegue Actual

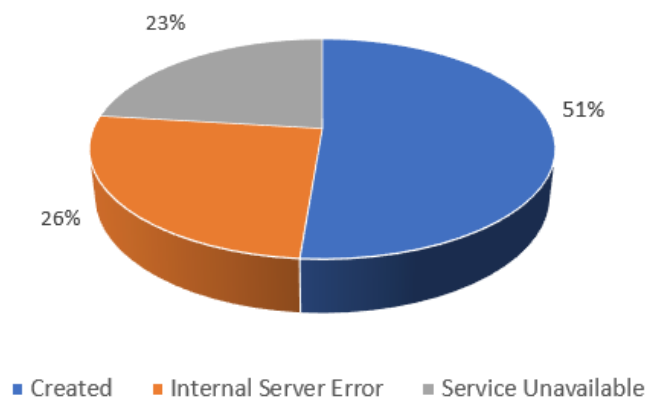
Porcentaje de Cumplimiento Requerimiento
Tiempo de Respuesta



Tipos de mensajes de respuestas obtenidos durante la prueba-

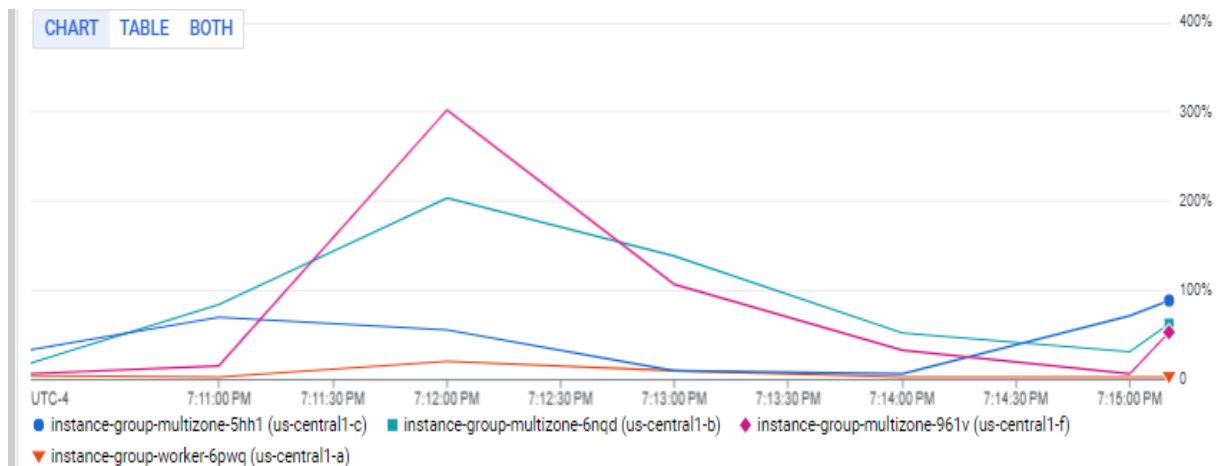
Despliegue Actual

Response Message



Análisis de Resultados Escenario 1

El comparativo del porcentaje de los mensajes de éxito recibidos por las peticiones realizadas durante la prueba para el escenario actual de despliegue aumentó en 7% lo que indicada que la estrategia de desplegar nodos sobre dos zonas de disponibilidad tuvo un efecto positivo en este aspecto. Sin embargo, ya aún un número significativo de peticiones presenta respuestas no satisfactorias se debe considerar el impacto de otros aspectos en el desempeño del sistema como lo es la utilización de CPU.



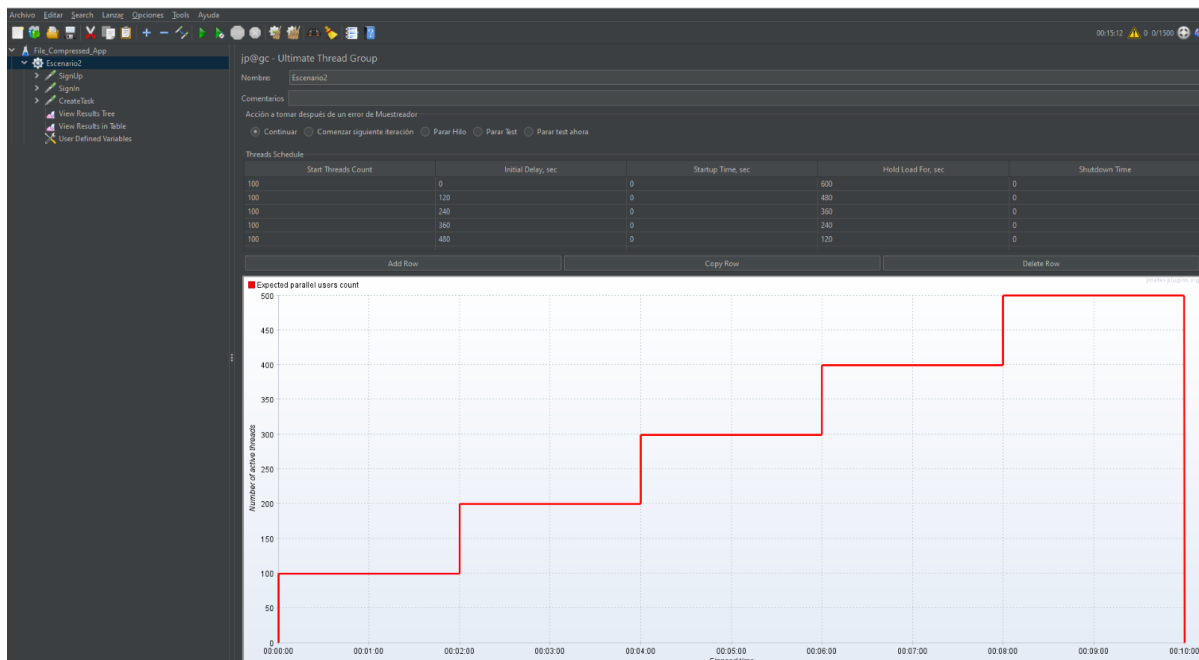
La gráfica anterior presenta la utilización de CPU de las instancias del web server durante la ejecución de la prueba. De manera particular, es posible ver que una de las instancias no logra responder a la carga enviada. Aunque la instancia en mención pasa el health check creado su desempeño no es favorable por lo que se recomienda incluir nuevos health checks que permitan identificar la causa raíz del bajo desempeño de dicha instancia.

Por otra parte, los resultados del porcentaje del requerimiento de tiempo de respuesta aumentaron significativamente pasando del 6% al 44% lo que se atribuye al despliegue de nodos en dos zonas de disponibilidad.

Aún existe un porcentaje significativo de peticiones con respuesta de tipo Internal Server Error y Service Unavailable (49% en total). Referente a esta situación se podría considerar el uso de más zonas de disponibilidad, al aumento del número máximo de instancias en la política de autoscaling y la reducción del valor de referencia utilizado en el load balancer.

Escenario 2. Se deberá definir un escenario donde se pueda probar cuál es la máxima cantidad de archivos que pueden ser procesados por minuto en la aplicación local. Para hacer pruebas de estrés se recomienda utilizar la herramienta Apache Bench (ab) o JMeter. Las pruebas de estrés deberán realizarse desde otro equipo diferente a los utilizados para ejecutar el servidor web y el servidor de base de datos. El escenario y los resultados de las pruebas de estrés deberán ser documentados con gráficas que ilustran cómo se comporta la aplicación a medida que el número de usuarios procesando archivos se incrementa, hasta llegar al punto en que el tiempo para iniciar el procesamiento de un archivo enviado por un usuario supere los 10 minutos (600 segundos). Restricciones del escenario. El archivo enviado a convertir durante las pruebas debe ser de un tamaño mínimo de 15 MiB y un máximo de 20 MiB.

Para la realización de esta prueba se utilizó un archivo de 20 MiB. En JMeter se creó una rampa que se muestra en la siguiente imagen:



Esta rampa tiene una duración total de 10 minutos.

A continuación, se presentan los resultados obtenidos para las pruebas realizadas en el Despliegue Previo (Bucket File Storage + Autoescalado + Load Balancing) y el Despliegue Actual (Alta disponibilidad del Web Server + Servicio de Mensajes Cloud Pub/Sub + Worker con Autoscaling (Entrega Actual)).

De acuerdo con los resultados presentados en la entrega anterior, para el Despliegue Previo no fue posible identificar un punto en el cual la aplicación tuviera un tiempo procesamiento mayor a 10 minutos ya que de las 802 peticiones de crear tarea realizadas solo 100 tareas fueron creadas de manera exitosa y al consultar la base de datos 8 minutos después de la ejecución de la prueba todas estaban en estado *Disponible*.

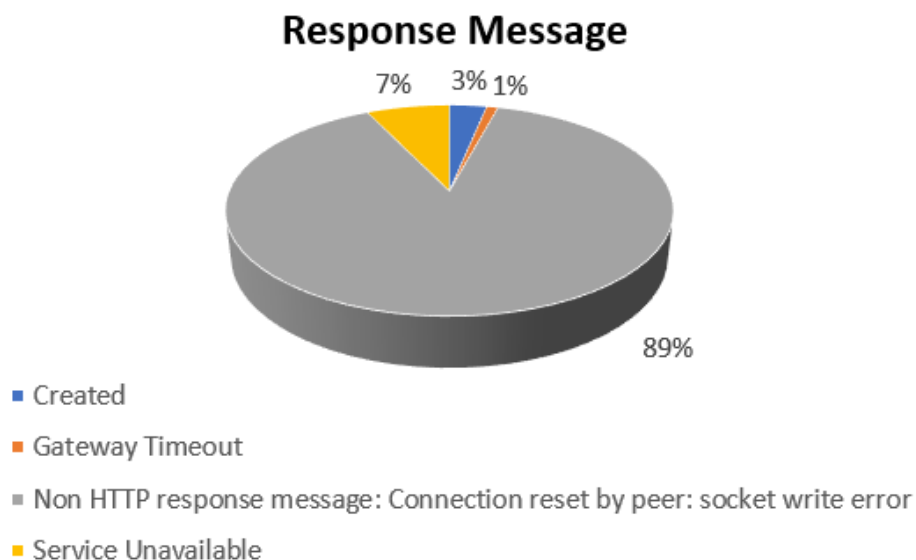
Para las pruebas de este escenario *Despliegue Actual* se realizó la misma metodología usada en la entrega anterior. Una vez se corrieron las pruebas en JMeter se esperó 10 minutos y se hizo una consulta a la base de datos para obtener todas las tareas creadas hasta el momento con su respectivo estado.

Dicha consulta se realizó a través de Postman usando en el endpoint de Tasks, la respuesta de la petición fue exportada en formato Json y posteriormente analizada en Excel donde se identificó que durante el transcurso de la prueba fueron creadas 13 tareas y todas se encontraban en estado *Uploaded*.

✕	✓	<i>fx</i>	= Origen{1368}	▼
nombre_archivo	ArchicoTets_Cloud.pptx			
usuario_id	1			
id	1371			
estado_tarea	Record			
usuario	1			
fecha_modificacion	null			
extension_final	Record			
estado_conversion	Record			
fecha_creacion	2023-05-14T23:22:31.618353			
llave	UPLOADED			
valor	1			

✕	✓	<i>fx</i>	= Origen{1355}	▼
nombre_archivo	ArchicoTets_Cloud.pptx			
usuario_id	1			
id	1358			
estado_tarea	Record			
usuario	1			
fecha_modificacion	null			
extension_final	Record			
estado_conversion	Record			
fecha_creacion	2023-05-14T23:15:29.878337			
llave	UPLOADED			
valor	1			

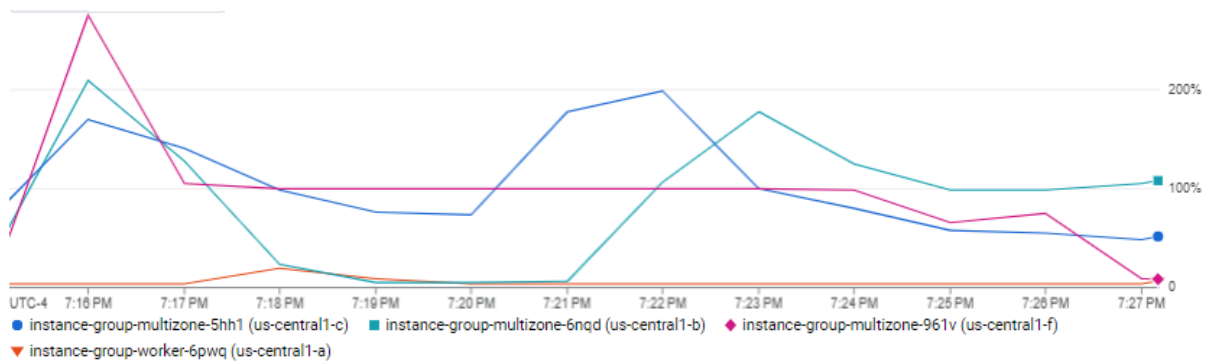
Durante la ejecución de la prueba del segundo escenario se enviaron al backend 405 peticiones de Crear Tarea registrando la distribución de respuestas mostradas en la siguiente imagen.



Análisis de Resultados

De acuerdo con los resultados obtenidos, un porcentaje considerable (>89%) de las peticiones generadas durante el escenario fallan. Esta falla esta asociada a una condición de estrés presentada por las instancias que se puede evidenciar en la siguiente imagen donde la utilización de CPU de las instancias es igual o mayor al 70% durante la mayor parte de la prueba.

CPU Utilization



Dada esta condición, es necesario aumentar la capacidad de las instancias utilizadas como web server para que puedan tolerar las cargas actuales consideradas en la rampa de carga de JMeter buscando de esta forma cumplir con los requerimientos esperados referente al tiempo de procesamiento de los archivos y la experiencia del cliente.

Conclusiones

- Se evidenció una mejora en los tiempos de respuesta registrados en el escenario 1. Esta mejora se atribuye al despliegue de nodos del web server en dos zonas de disponibilidad.
- Debido a que en los escenarios aún un número significativo de peticiones presenta respuestas no satisfactorias se debe considerar el impacto de otros aspectos en el desempeño del sistema como lo es la utilización de CPU en futuras estrategias para la mejora del desempeño. Para efectos de este proyecto no se han realizado cambios en el tipo de instancias utilizadas debido a la restricción en los créditos disponibles.
- Otras posibles opciones para la mejora del desempeño a considerar son: uso de más zonas de disponibilidad, al aumento del número máximo de instancias en la política de autoscaling y la reducción del valor de referencia de utilización de CPU utilizado en el load balancer.
- Se debe considerar la posibilidad de incluir parámetros adicionales en la valoración del desempeño de las instancias ya que se ha identificado que en algunas ocasiones aunque la instancia pasa el health check no está presentando el desempeño esperado.