

Cloud conversión tool – Escenario 2

A continuación, encontrará información sobre como configurar y correr el escenario 2 de pruebas.

1. Pre-requisitos:

- Deben existir y estar funcionando por lo menos las instancias de VM monitoring-worker y worker.
- Debe existir la instancia db1 de Cloud SQL.
- El repositorio debe estar clonado en Cloud Shell.
- Debe existir por lo menos un usuario registrado en la base de datos: se puede hacer una petición a `/api/auth/signup` por medio de postman para lograr esto.

2. Componentes:

a. **trigger**: Disparador encargado de enviar tareas a la cola de tareas.

- Variables importantes:
 - NUM_PARALLEL_TASKS: Cantidad de peticiones paralelas a correr
 - NUM_CYCLES: Cantidad de veces que se quiere mandar las peticiones paralelas.
 $\text{NUM_PARALLEL_TASKS} * \text{NUM_CYCLES} = \text{total peticiones.}$
 - OLD_FORMAT: Formato original del video a utilizar.
 - NEW_FORMAT: Formato a convertir el video
 - DEMO_VIDEO = Video a utilizar para el experimento.

b. **monitor**: Encargado de monitorear diferentes eventos que dispara una tarea y sacar las métricas solicitadas.

- Eventos:
 - TASK SENT: Cuando la tarea fue enviada a la cola de tareas.
 - TASK RECEIVED: Cuando la tarea fue recibida.
 - TASK STARTED: Cuando la tarea comenzó su procesamiento.
 - TASK SUCCEEDED: Cuando la tarea fue finalizada con éxito.

3. Pasos para correr el escenario:

a. Copiar las carpetas trigger y monitor dentro de la instancia monitoring-worker, para eso ubicarse en MISW4204/gcp y poner los siguientes comandos:

```
cd MISW4204/gcp/
```

```
export ZONE=us-central1-c
```

```
gcloud compute scp --recurse ../monitor monitoring-worker:/tmp/monitor  
--zone $ZONE
```

```
gcloud compute scp --recurse ../trigger monitoring-worker:/tmp/trigger
--zone $ZONE
```

- b. Entrar con SSH a la instancia de monitoring-worker, y cambiar las variables a conveniencia en el archivo /etc/api.env.
- c. Dentro de la instancia de monitoring-worker, construir la imagen y el contenedor del monitoring, para eso correr con sudo su los siguientes comandos:

```
cd /tmp/monitor

docker build -t monitoring-image .

docker run -d --env-file /etc/api.env --name monitoring-container
monitoring-image

docker logs -f monitoring-container
```

- d. En otra terminal, entrar con SSH a la instancia de monitoring-worker, construir la imagen y el contenedor del trigger y correr con sudo su los siguientes comandos (Es importante que antes de hacer eso se asegure que el contenedor del monitor está corriendo correctamente):

```
cd /tmp/trigger

docker build -t trigger-image .

docker run -d --env-file /etc/api.env --name trigger-container -v
/mnt/video:/video trigger-image

docker logs -f trigger-container
```

- e. Lo anterior empezará a correr el experimento. Esperar a que termine, cuando eso pase, la terminal donde se está corriendo el contenedor del monitor le debe arrojar las métricas de los resultados del experimento, algo por el estilo:

```
-----
```

Reporte

Total peticiones: 10

Peticiones concurrentes: 5

Tiempo de respuesta por petición promedio (ms): 411499.30

Tiempo de respuesta (ms) P95: 709008.52

Peticiones por minuto (Throughput): 0.81

- f. Para volver a correr el experimento, borrar los contenedores y volverlos a correr.