

Pruebas de Rendimiento del Proyecto Opcional

Andrés Bonilla Solano - 2023101220
Luis Fernando Benavides Villegas - 2023072689
Juan Diego Jiménez - 2019199111
Alex Naranjo - 2023063599

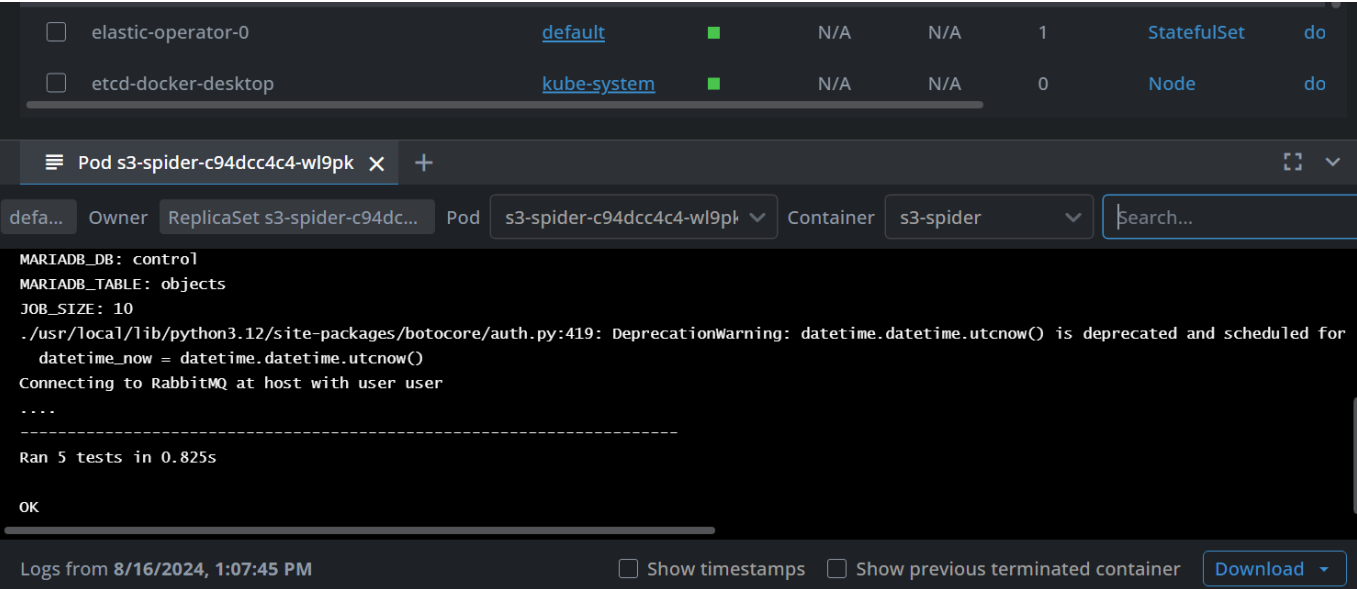
Pruebas realizadas:

Revisión de pods para verificar si están funcionando. Para las pruebas unitarias solo hay que ejecutar el proyecto y luego revisar en los pod logs si las pruebas tuvieron exito. Esto se verá parte por parte en el siguiente apartado donde se mostraran los resultados de las pruebas unitarias.

Resultados de las pruebas unitarias:

Para las pruebas unitarias, se utilizó la librería de unittest que viene incluida en las versiones más recientes de python. En caso de que las pruebas sean existosas en los logs del pods debería aparecer "RAN {numero_de_tests} in {tiempo_de_ejecución}" y luego un "OK". En caso contrario en lugar del "OK", aparece FAILED (failures={numero_de_fails}) y indica en que linea falló y que tipo de error ocurrió.

En la siguiente imagen, está el resultado de las pruebas unitarias del s3-spider. Para esta parte del proyecto, se usaron cinco funciones principales: **list_s3_objects**, **download_s3_object**, **create_jobs**, **save_jobs_to_db**, **publish_jobs_to_rabbitmq**. Para estas funciones se creó un entorno simulado para el bucket y también se utilizaron datos de prueba que se asemejan a los del proyecto. Asimismo, se utilizó la librería **moto** para simular los servicios de AWS y el módulo mock de unit test para simular objetos y partes de las funciones.



En el caso del downloader, se intentó hacer un unit test pero había un elemento de la función de procesar mensajes que no se podía imitar utilizando las librerías mencionadas anteriormente. Al intentar hacer el unit test da el siguiente error: pika.exceptions.AMQPConnectionError, esta excepción es de la librería de Pika, la cual se esta utilizando para interactuar con RabbitMQ. Esta excepción indica que ocurrió un error al intentar establecer la conexión con el servidor. Pese a que se intentó sustituir el servicio temporalmente usando `@patch('pika.BlockingConnection')`, el error seguía apareciendo. Sin embargo al ejecutar el proyecto, en los pods logs aparece que se estan esperando mensajes por lo que se garantizó que esta funcionando

correctamente.

Para el spark-job no se hicieron pruebas unitarias como tal ya que es un archivo .scala.