ClearBlade Challenge

Jakub Murici

- Program overview:

  The program that I created scans user's current CPU at a certain interval using the 'psutil' Python library. I hardcoded the default setting to have a CPU scan every **two** seconds for **two and half minutes** which I believe is the optimal ratio to really watch the program do what it is supposed to do. The duration of the scanning and the gap between each scan can of course be changed inside the python file.

  The program sends the CPU usage value to a topic called 'CPU-Usage' on the ClearBlade platform and the value is immediately saved to a collection called 'challenge' every time the messaging client receives a scan. The data analysis part of the project calculates an average CPU usage over a period of time and saves the results to the messages to a topic called 'analytics'. Based on the sample scenario (two and half minutes, scan every two seconds), the program works the following way. Every minute the program calculates the sum of CPU usage for the **past minute** and divides the value by the number of scans to get the average. This average is saved to 'analytics' topic **every minute.** The data from the 'challenge' collection is then **completely erased** right after the CPU average is published so that way, we get a CPU average performance for the next minute again and not for two minutes. If the collection was not erased, the CPU average would not be as accurate as we want it to be. If the python file is not scanning CPU, the program sends a notification to the 'analytics' topic that CPU average is not currently available, this also happens every minute.

  Please note that these trigger times and settings is python file are created for the best demonstration of the program, not for the best performance. I also tried to use a lot of comments, especially in the 'Average-CPU' code where the majority of the whole process happens to help understand the process better. I am also aware of a little bug in my system that I am still trying to overcome. However, this bug does not affect calculating the accurate results in any way. I noticed that although the CPU performance is sent to the messages once for each scan, it is saved twice into the collection. I tried looping through the 'waitingformessage' only once, but it still saves it twice. It seems that the code is triggered twice whenever it catches a message regardless of using Stream service or Micro service that triggers on publish. Since I am calculating an average, it does not matter since it's added twice for each scan, but if I wanted to perform a different analysis, this could cause some issues. I am still looking into it and I'll update it once I find out what is causing this problem.

- Tutorial:
    1. Clone ‘JakubMurici/CPU-Calculator’ into your python IDE
    2. Clone JakubMurici/Internship-Challenge into your ClearBlade Platform.
    3. Copy the generated SystemKey and SystemSecret into the challenge.py file.
    4. Run the challenge.py file, you will be asked to create an account or login (in case you created an account inside the platform).
    5. After accessing the platform, please let the scan run for full 2.5 minutes to get the best results for the first time. Feel free to adjust it afterwards.
    6. Check the ‘analytics’ to find out what was your CPU performance over the given time.

- Final thoughts:

    I really enjoyed working with the platform. It seemed somewhat hard in the beginning, it took me some time to get familiar with the platform and how to work with it properly but once I figured out the basics such as connecting to the platform/messages, it turned out to be fairly easy. Now looking backwards at what I have done, the process seems pretty straightforward. What helped me the most is the fact that the Docs for both, python SDK and the JavaScript on the actual platform are extremely well written. I haven’t explored all the available systems yet but ClearBlade Platform seems as something that you can do a lot of different and GREAT analysis with. It was a good challenge!!!