DataEng: Data Maintenance In-class Assignment

This week you will gain hands-on experience with Data Maintenance by constructing an automated data archiver that compresses, encrypts and stores pipelined data into a low-cost, high-capacity GCP Storage Bucket.

Submit: Make a copy of this document and use it to record your results. Store a PDF copy of the document in your git repository along with any needed code before submitting for this week.

Your job is to develop a new, separate python Kafka consumer similar to the consumers that you have created multiple times for this class. This new consumer should be called archive.py and should receive all data from a test Kafka topic, compress the data, encrypt the data (optional) and store the compressed data in a GCP Storage Bucket.

Note that each member of your team should build their own archive mechanism. As always, it is OK to help one another, but each person should develop their own python program for archiving.

Discussion Question for Your Entire Group (do this first)

When archiving data for a data pipeline we could (a) compress, (b) encrypt and/or (c) reduce the data. Here, “reducing the data” refers to the process of transforming detailed data, such as 5 second breadcrumbs for all buses on all trips, into coarser data that contains, for example, only contains a subset of the original data such as only some buses, some trips or possibly fewer breadcrumbs per trip.

Under what circumstances might each of these transformations (compress, encrypt, reduce) be desirable for a data archival feature?

Chinmay Tawde - One circumstance is where we have limited data storage, in that case compression helps reduce the data size considerably, and thus save a lot of space. Similarly reduce would also help since having a coarser data would definitely lead to less file size compared to the original file size. Encryption on the other hand in this scenario depends, as some encryption can make the data take up more space instead due to the algorithm it uses to encrypt the data. So encryption is not ideal in this situation.

Deepa Hegde - Agrees to what I said

Varun Jaisundar Raju - Agrees to what I said

A. Create test topic

Create new Kafka producer and consumer programs as you did with the Data Transport in-class activity (link to Transport activity). Create a new Kafka topic that is separate from the topic(s) used for your project. Call it “archivetest” or something similar. As with the Data Transport activity you should initially have your new producer produce test data and have a single consumer that consumes any/all data sent to the Kafka topic.

B. Create separate consumer groups

Similar to part H in the Transport activity, create two separate Consumer Groups for your new Kafka topic. Run a separate consumer for each group and verify that each consumer consumes all of the data sent by the producer.

Your first consumer should simply print all data that it receives. This consumer simulates the main branch of your data pipeline. Typically, the main branch of your pipeline would validate, transform, integrate, load, etc. the data, but for this in-class assignment it only needs to print the data.

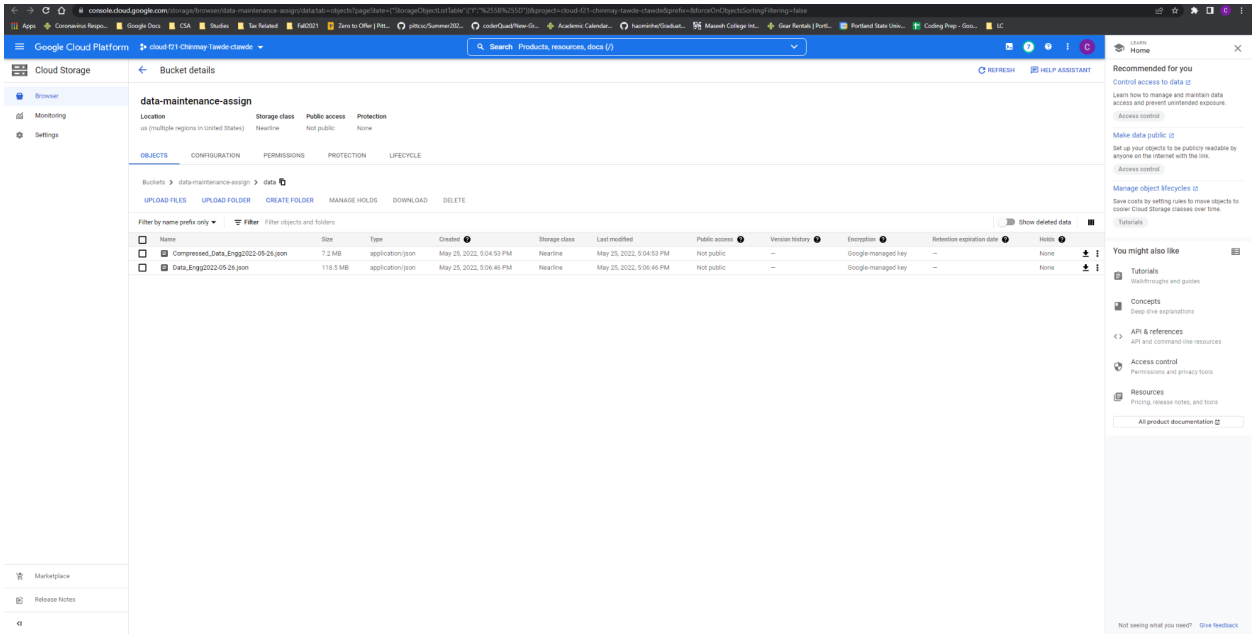
C. Archive the data in a GCP Storage Bucket

Your second consumer (call it archive.py) should store all received data into a GCP Storage Bucket. You will need to create and configure a Storage Bucket for this purpose. You are free to choose any of the available storage classes. We recommend using the Nearline Storage class.

D. Compress

Modify your archive.py program to compress the data before it stores the data to the storage bucket. Use zlib compression which is provided by default by python. How large is the archived data compared to the original?

- The archived data is only 7.2 MB whereas the original data is 118.5 MB



E. Encrypt (Optional)

Next, modify your archive.py to encrypt the data prior to writing it to the Storage Bucket. Your archive.py program should encrypt after compressing the data. Use RSA encryption as described here: link There is no need to manage your private encryption keys securely for this assignment, and you may keep your private key in a file or within your python code.

Be sure to test your archiver by decrypting and decompressing the data stored in the Storage Bucket. We suggest that you create a separate python program for this purpose.

Now large is the archived data?

F. Add Archiving to your class project (Optional)

Add your archive.py program (or something like it) to your class project's pipeline(s). Mention this in your final project presentation video. Because your project is shared among your project team members you will need to coordinate the adding of new Kafka consumer groups so that each team member may safely add their own archiving service. Again, it is not necessary to securely manage your RSA private encryption key, and it is OK to keep it in a file or in your python source code.

G. Virtual Machine Resource Usage Monitoring

This section covers how to monitor your virtual machine's health. Walk through this activity together with your project group since you have the same project VM. In particular we'll be

looking into how to check the resource usage on your host. As this is a virtual machine, you have the choice of using traditional Linux command line tools like top, or using GCP UI tools.

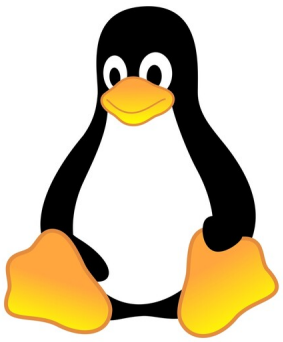
For the questions about the pipeline, you can either use the pipeline from the activity above, or modify your course project pipeline. Be sure to set `autocommit = False` when setting up the database connection so it doesn't actually commit the data to your tables and mess up your data. Use the new endpoint: http://www.psudataeng.com:8000/getBreadCrumbDataV2 This endpoint just serves the data from the start of the course again. It'll be shut down at the end of day Friday.

Some handy links are listed below. We recommend skimming or searching them as a reference during the rest of this activity. You are not required to read them. Remember to Google anything that isn't immediately clear!

- Linux command line monitoring tools: https://www.makeuseof.com/best-cli-tools-to-monitor-linux-performance-terminal/
    - My personal favorites: dstat, top, free -h, df -h, du -h -d1, ps aux
- GCP Monitoring overview: https://cloud.google.com/monitoring/docs/monitoring-overview
- GCP VM monitoring dashboards: https://console.cloud.google.com/monitoring/dashboards/resourceList/gce_instance?tab=overview
- GCP Monitoring Processes: https://cloud.google.com/monitoring/agent/process-metrics
- GCP Alerts doc: https://cloud.google.com/monitoring/monitor-compute-engine-virtual-machine
- GCP Alerts UI: https://console.cloud.google.com/monitoring/alerting?walkthrough_id=monitoring_quickstart_compute_uptime

Do you have any other monitoring tools you prefer?

H. Getting to know your VM



Which Linux distribution are you using? (Distro and year: Ubuntu 20)

Debian, Debian GNU/Linux, 11 (bullseye)

What are your system's resource allocations? How much of each of the following resources do you have on the VM?

- CPUs
- CPU cores

```
ctawde@data-storage-assign:~$ lscpu | egrep 'Model name|Socket|Thread|NUMA|CPU\(s\)'
CPU(s):                          2
On-line CPU(s) list:             0,1
Thread(s) per core:              2
Socket(s):                       1
NUMA node(s):                    1
Model name:                      Intel(R) Xeon(R) CPU @ 2.20GHz
NUMA node0 CPU(s):               0,1
```

- Memory
  4 GB

```
ctawde@data-storage-assign:~$ grep MemTotal /proc/meminfo
MemTotal:        4024888 kB
```

- Swap space

```
ctawde@data-storage-assign:~$ grep Swap /proc/meminfo
SwapCached:            0 kB
SwapTotal:             0 kB
SwapFree:              0 kB
```

- Disk

```
ctawde@data-storage-assign:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            2.0G     0  2.0G   0% /dev
tmpfs           394M  412K  393M   1% /run
/dev/sda1       9.7G  4.2G  5.0G  46% /
tmpfs           2.0G   16K  2.0G   1% /dev/shm
tmpfs           5.0M     0  5.0M   0% /run/lock
/dev/sda15      124M  5.9M  118M   5% /boot/efi
tmpfs           394M     0  394M   0% /run/user/1001
```

I have overall 10 GB of disk space on my sda1 partition and I have used around 4.2 GB out of it.

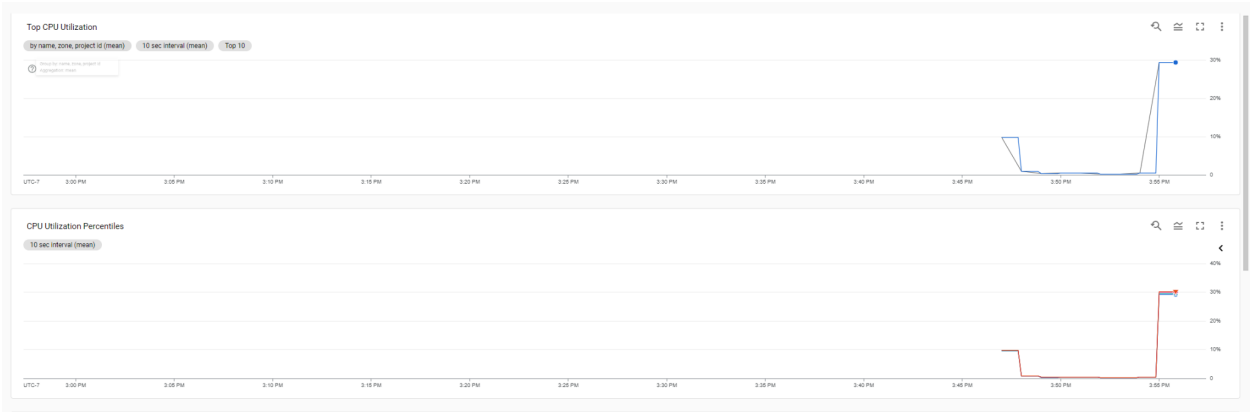
Are there other resources you're used to monitoring? When are those resources important to keep an eye on?

Ans: If you have a limited network connection (such a limited download limit/ upload limit) then you can monitor that as well.

I. Pipeline Resource usage

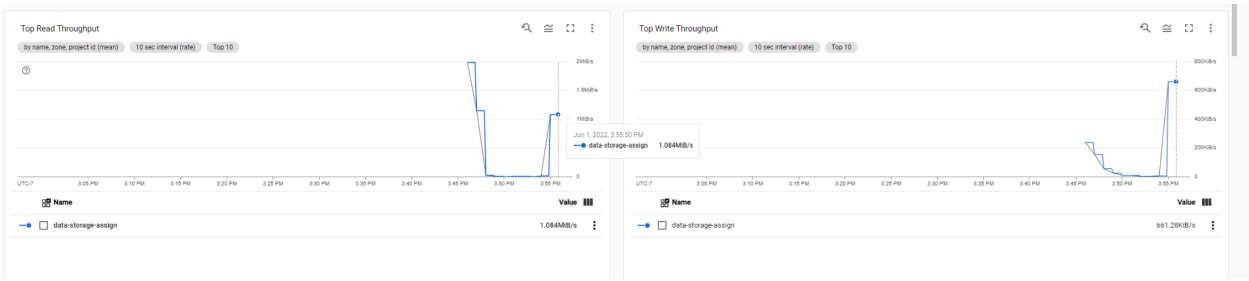
When your pipeline is running, what is the typical system-wide usage of these resources:

- CPU

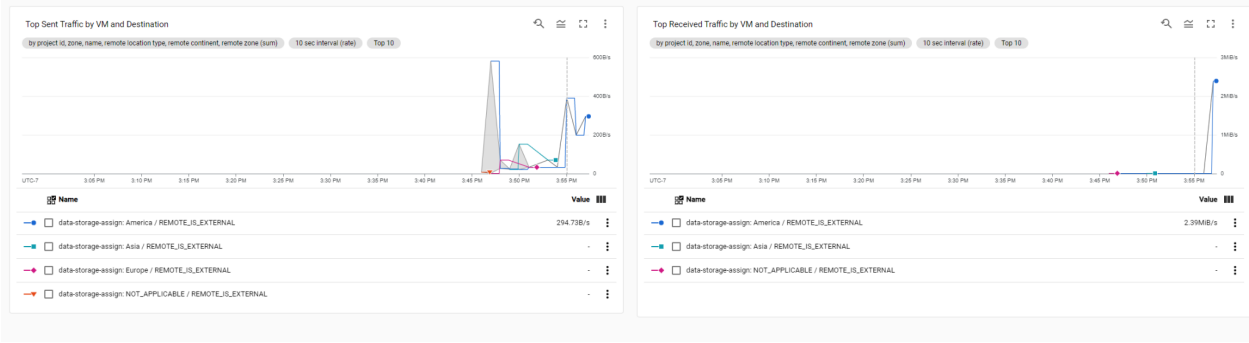


CPU usage is around 30% while running the pipeline

- Load Average
- Memory
- Disk Usage (fullness)
- Disk Utilization (I/O)



- Network (I/O)



When your pipeline is running, what are some of the top processes that are running? Which ones are consuming the most CPU and memory?

```
ctawde@data-storage-assign:~$ ps -eo pid,ppid,cmd,%mem,%cpu --sort=-%mem | head
    PID    PPID CMD                         %MEM %CPU
   4673    4639 python ./producer_v2.py -f  17.8  8.3
   2724       1 /usr/lib/postgresql/13/bin/  0.7  0.0
    404       1 /usr/bin/google_osconfig_ag  0.6  0.0
    403       1 /usr/bin/google_guest_agent  0.4  0.0
    409       1 /usr/bin/python3 /usr/share  0.4  0.0
    190       1 /lib/systemd/systemd-journa  0.3  0.0
   2728    2724 postgres: 13/main: walwrite  0.2  0.0
      1       0 /sbin/init                   0.2  0.0
   2729    2724 postgres: 13/main: autovacu  0.2  0.0
```

Ans: The python program is the top process which is running utilizing about 17.8% of memory and 8.3% of CPU

J. Understanding Linux Monitoring Metrics

What is the difference between disk usage and disk utilization?
Disk usage is the portion or percentage of computer storage that is currently in use.
Whereas Disk utilization is the graph that shows current disk utilization while performing some I/O operations.

What is the difference between memory that is free and memory that is available, in the command `free -h`? Is it concerning when there is very little “free” memory, if there is a lot of “available” memory?

```
gen@dataeng:~$ free -h
               total        used        free      shared  buff/cache   available
Mem:           968Mi       189Mi        79Mi        12Mi       699Mi       624Mi
Swap:             0B          0B          0B
```

Ans: Free memory is memory that is being wasted and not used at all whereas available memory is memory which is available for allocation to new or existing processes. Available memory is then an estimation of how much memory is available for use without swapping.

Does swap space use memory or disk? Is it a good or bad sign (or both/neither) when it is used? What is a standard amount of swap pace to allocate?
Swap space uses disk space when memory is insufficient. According to me It is neither good nor bad as it just helps the system to use other available resources when memory is not enough.

Should you be concerned if the load average is 7?
Ans: It depends on the number of CPUs, so the load average is 7 for a single CPU system, we should be concerned but if it was 7 for 8 core CPU that it is still underutilized and there is nothing to worry about.

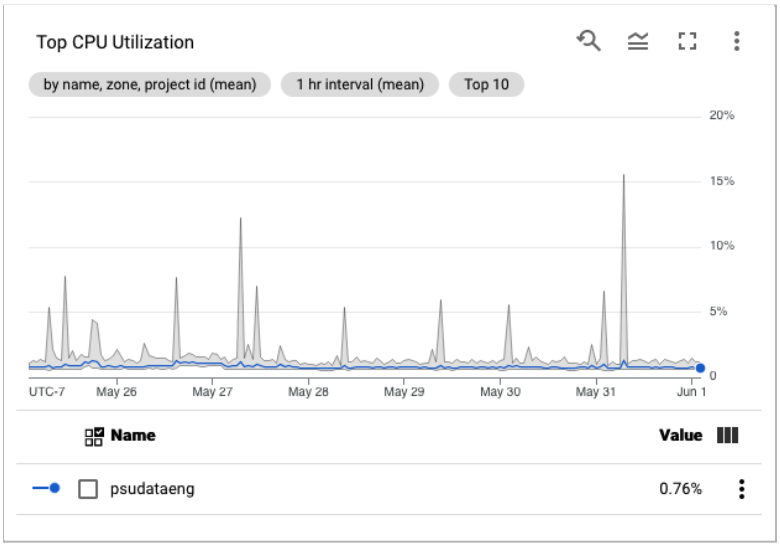
Yes, the previous question is a trick question. The answer to your followup question: the system has 32 cores. Are you concerned now?

Ans: In a 32 core CPU, we don't need to be concerned at all as the system is underloaded by 3 times at least and can handle load average uptil 32 to be utilized at 100%.

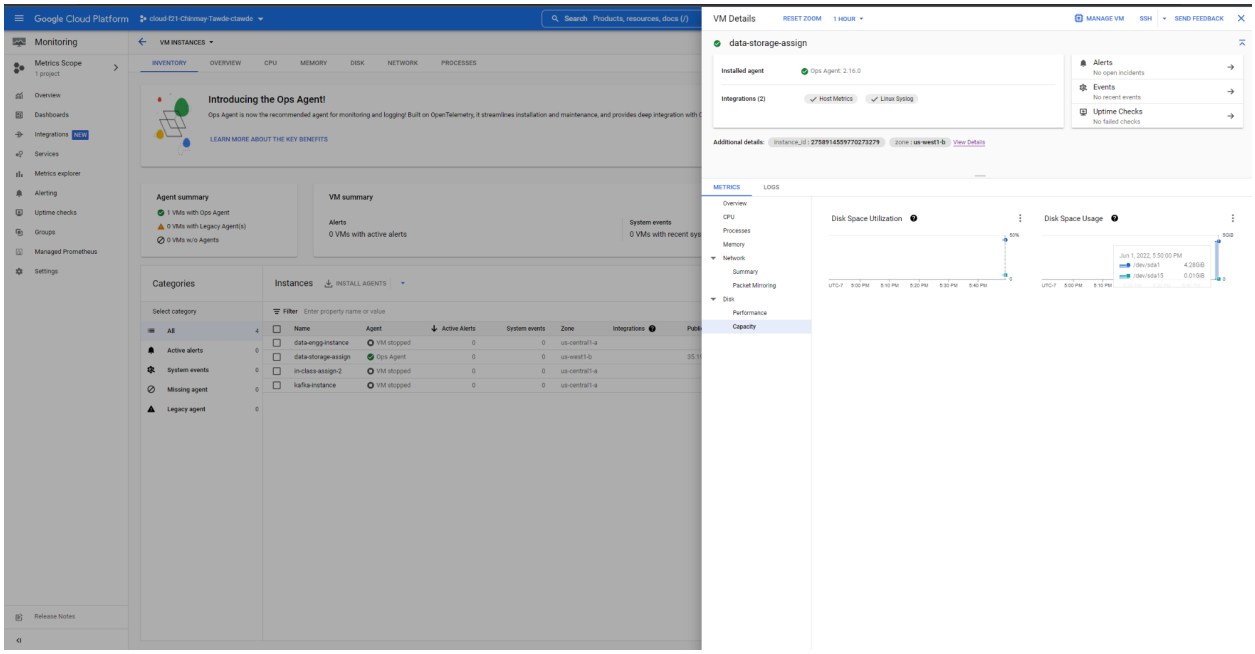
Should you be concerned if the 1 minute load average is really high, but the 5 and 15 minute load averages are normal? Why or why not? What does this mean, is the load going up or down?
Ans: I don't think there's a reason to be concerned as it only means resources are being utilized or allocated properly as needed over time and yes, it means the load is fluctuating.

K. GCP Monitoring

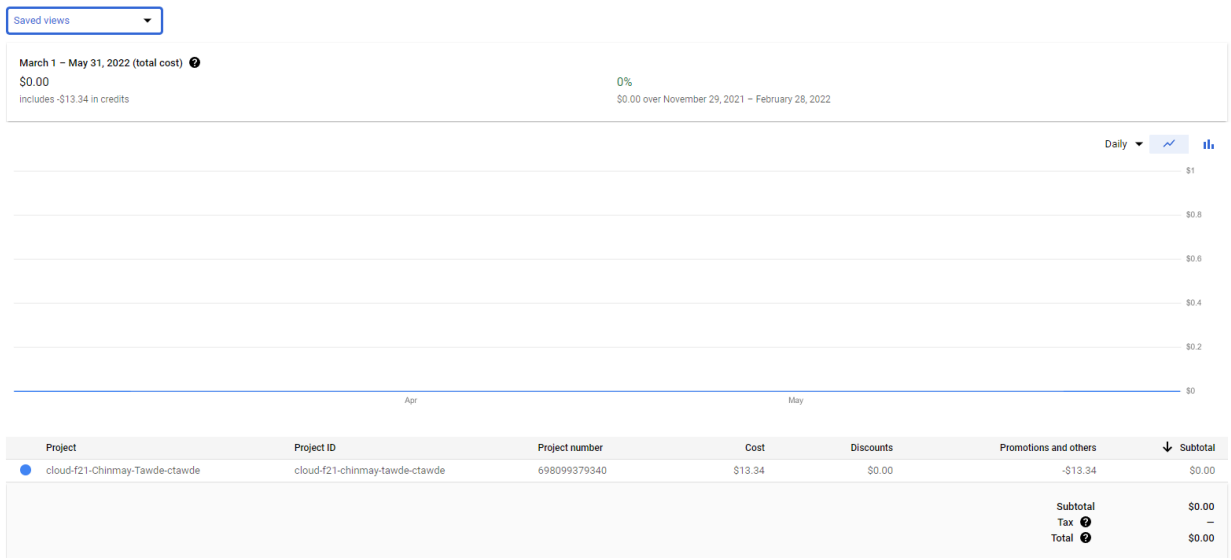
Where in the GCP cloud console can you see charts like the one below, showing the resource usage of your VM?
Ans: In The monitoring section under dashboards you can find charts like below, https://console.cloud.google.com/monitoring/dashboards/resourceList/gce_instance?tab=overview



Where in the GCP cloud console can you see disk usage (not utilization)?
Ans: After installing the Ops agent through the monitoring dashboard you can check individual VM related statistics such as this:



How can you see GCP billing usage, excluding credits from promotions and course coupons?
Ans: You can go to billing and click on Reports section to find details related to GCP billing usage



L. Predicting and Preventing Outages

Consider the following questions, and how you can improve the resiliency of your VM and pipeline.

Running out of resources

Judging from your current usage patterns, and if nothing were to change, on what date do you expect to run out of disk space? What about your file archive GCP storage bucket? When will you run out of GCP credits?

A picture of a full disk on Windows. Linux just doesn't run out of disk space with quite the same flamboyance.

When your pipeline is downloading new data, is the system running out of any resources? Did it run out of memory? Does it start paging/swapping? Was the CPU pegged? What was the normalized CPU usage (total CPU usage, averaged over the count of CPU cores)?
Ans: As we continuously keep fetching new data and storing it onto disk, the disk keeps getting full. So clearing old data to free up disk space is one of the main issues.
On a small machine it goes out of memory but a e2-medium instance handles the load well.

Judging from the last question, which resource would you say is the biggest bottleneck?

Ans: The limited disk space is the bottleneck as we have incoming data on a daily basis.

Alerting on and resolving problems

Given the typical resource usage patterns we've explored here, if you were to create some resource usage alerts, what types of thresholds would you set? For example, would you get out of bed to fix a disk at 70% full or wait till 90% full? Thinking of typical human behavior, what issues can you think of if alert thresholds are set too high or too low?

Ans: I have worked as a Systems engineer at PSU in OIT wherein I have seen alerts being set when a machine uses 90% of CPU resources. Alerts when a high amount of network traffic is flown through a VM, Disk usage reaches 80%, etc.

If your system were running out of disk and you needed to free up space right away, what files would you delete first? Could you recreate the data from the remaining files? Do you have a safe copy of the data elsewhere?
Ans: The best way to clean up space is to delete temporary files present on the system. Usually on windows, we have a folder called “temp” which stores snapshots, some data when you unzip files, browser history and cache, etc. Clearing that can free up around 2 - 5 Gb of space!
Other than that, you can try clearing older files by date depending on business requirements. Or start uploading them on a cloud environment somewhere instead.

If your system were running out of memory, and you knew it was caused by your consumer, how can you kill the consumer process? What effect (if any) would this have on the integrity of data in your database? Could you safely start the consumer again once the system is healthy again, or would you need to fix any data first? Can you think of any database features that could make it more resilient to failures in the middle of running the pipeline?

Ans: The process that runs the python program can be killed, the data integrity depends on the commit strategy used. If we commit at the end of the complete pipeline. Then if we kill the process in between, nothing would be committed. Thus, the database would still be in a clean state with no change and we can start our consumer again right away. But if we commit at regular intervals, then we might have to revert or delete some records using queries.