**Here is a link to my code: https://github.com/NAlexH2/de-proj-cs510**

**DataEng Project Assignment 1 Submission Document**

Construct a table showing each day for which your pipeline successfully, automatically processed one complete day's worth of sensor readings.

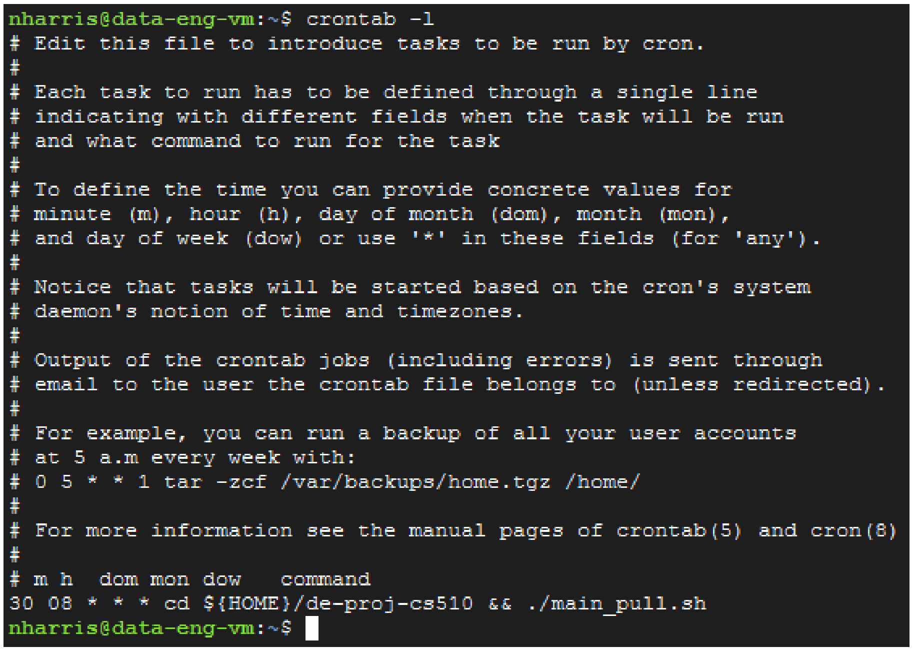
**Note from Alex:** I am understanding that the sensor readings from the FAQ is each individual event that took place for all vehicles summed together. This is why the pub/sub messages published and received are the same number, published every breadcrumb, received every published breadcrumb.

| Date | Day of Week | Approximate Time of day for your data access | # Sensor Readings | Total Data Saved (KBs) | # Pub/Sub messages published and received |
|---|---|---|---|---|---|
| 04/11 | Thursday | 5:00 PM | 314,815 | 105,808 | 0 |
| 04/12 | Friday | 5:00 PM | 295,348 | 99,253 | 0 |
| 04/13 | Saturday | 8:00 AM | 332,888 | 111,874 | 0 |
| 04/14 | Sunday | 8:00 AM | 342,189 | 114,995 | 0 |
| 04/15 | Monday | 8:00 AM | 328,164 | 110,298 | 0 |
| 04/16 | Tuesday | 8:00 AM | 235,787 | 79,236 | 0 |
| 04/17 | Wednesday | 8:00 AM | 262,890 | 88,350 | 0 |
| 04/18 | Thursday | 8:00 AM | 328,945 | 110,551 | 0 |
| 04/19 | Friday | 8:00 AM | 342,744 | 115,195 | 0 |
| 04/20 | Saturday | 8:00 AM | 340,480 | 114,438 | 340,480 |
| 04/21 | Sunday | 8:00 AM | 342,828 | 115,232 | 342,828 |

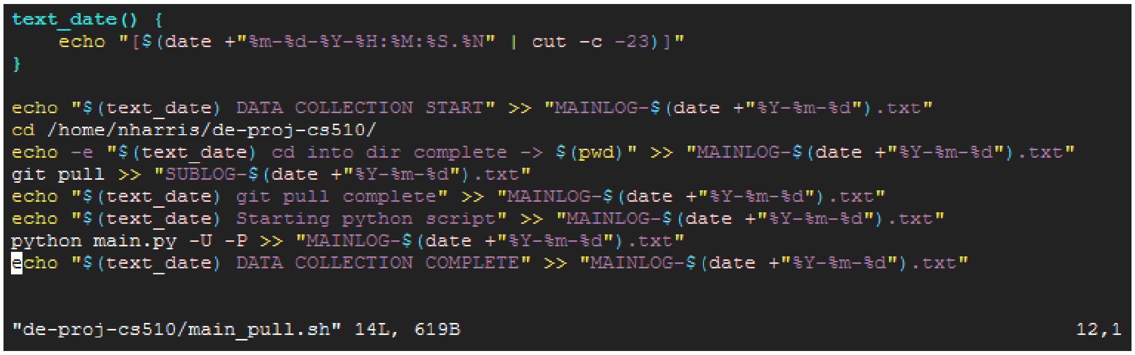
Additionally, include screenshots for the parts C, H and I

1. Output of crontab -l: Your scheduled cron jobs.

**Note from Alex:** Found it easier to just use a bash script to output data as well with. I'll be able to check each log daily. I wanted to do this for the subscriber, but the requirement of it using systemd makes it harder to set up. I may try and find a way to make it happen though.



```
nharris@data-eng-vm:~$ crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
30 08 * * * cd ${HOME}/de-proj-cs510 && ./main_pull.sh
nharris@data-eng-vm:~$
```

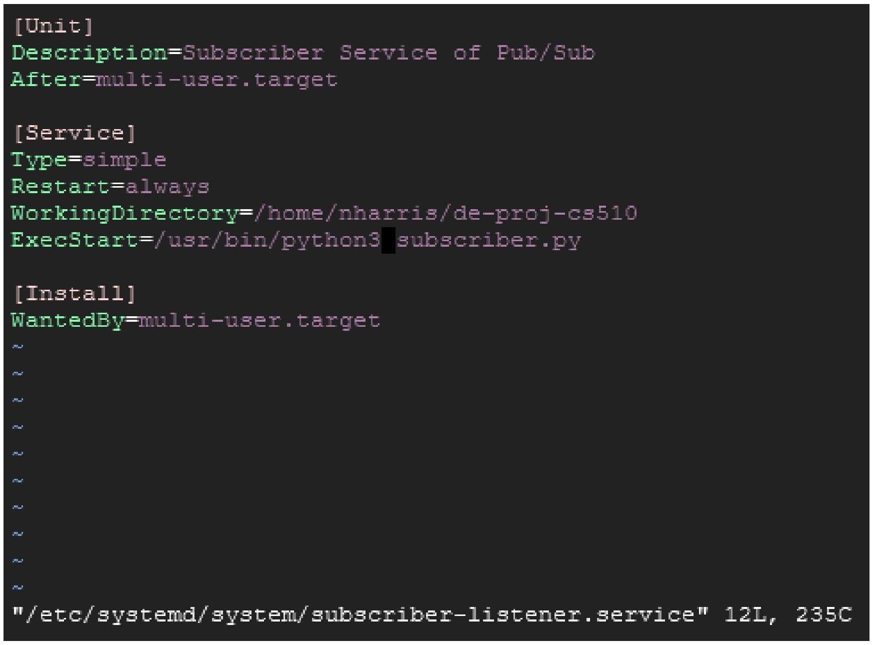


```
text_date() {
    echo "[$(date +"%m-%d-%Y-%H:%M:%S.%N" | cut -c -23)]"
}

echo "$(text_date) DATA COLLECTION START" >> "MAINLOG-$(date +"%Y-%m-%d").txt"
cd /home/nharris/de-proj-cs510/
echo -e "$(text_date) cd into dir complete -> $(pwd)" >> "MAINLOG-$(date +"%Y-%m-%d").txt"
git pull >> "SUBLOG-$(date +"%Y-%m-%d").txt"
echo "$(text_date) git pull complete" >> "MAINLOG-$(date +"%Y-%m-%d").txt"
echo "$(text_date) Starting python script" >> "MAINLOG-$(date +"%Y-%m-%d").txt"
python main.py -U -P >> "MAINLOG-$(date +"%Y-%m-%d").txt"
echo "$(text_date) DATA COLLECTION COMPLETE" >> "MAINLOG-$(date +"%Y-%m-%d").txt"


"de-proj-cs510/main_pull.sh" 14L, 619B                                    12,1
```

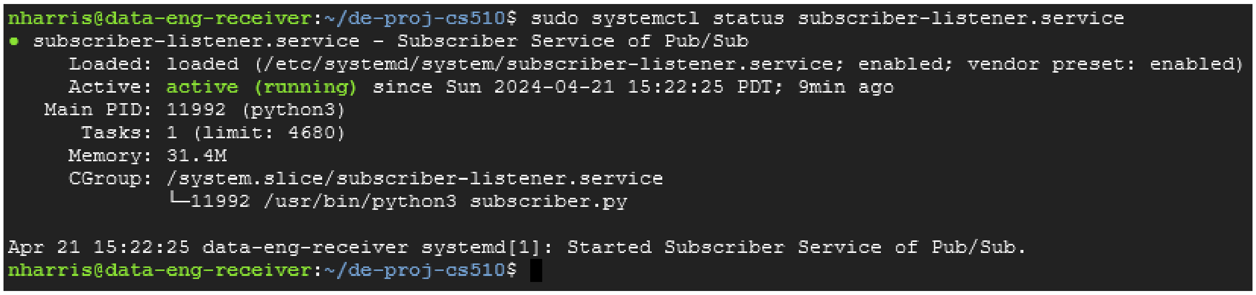
2. systemctl status: This will show the status of your receiver program.



```
[Unit]
Description=Subscriber Service of Pub/Sub
After=multi-user.target

[Service]
Type=simple
Restart=always
WorkingDirectory=/home/nharris/de-proj-cs510
ExecStart=/usr/bin/python3 subscriber.py

[Install]
WantedBy=multi-user.target
~
~
~
~
~
~
~
~
~
~
"/etc/systemd/system/subscriber-listener.service" 12L, 235C
```

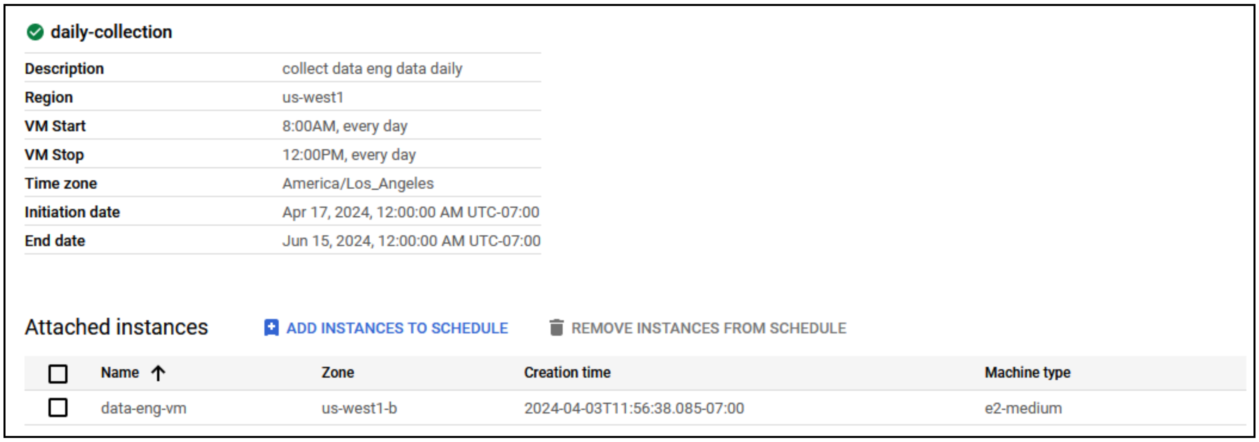


```
nharris@data-eng-receiver:~/de-proj-cs510$ sudo systemctl status subscriber-listener.service
● subscriber-listener.service - Subscriber Service of Pub/Sub
     Loaded: loaded (/etc/systemd/system/subscriber-listener.service; enabled; vendor preset: enabled)
     Active: active (running) since Sun 2024-04-21 15:22:25 PDT; 9min ago
   Main PID: 11992 (python3)
      Tasks: 1 (limit: 4680)
     Memory: 31.4M
     CGroup: /system.slice/subscriber-listener.service
             └─11992 /usr/bin/python3 subscriber.py

Apr 21 15:22:25 data-eng-receiver systemd[1]: Started Subscriber Service of Pub/Sub.
nharris@data-eng-receiver:~/de-proj-cs510$
```

3. VM instance schedule: This will display the schedule settings for your GCP VM instance.

**Note from Alex:** One is for the collector/publisher, the other is the receiver.

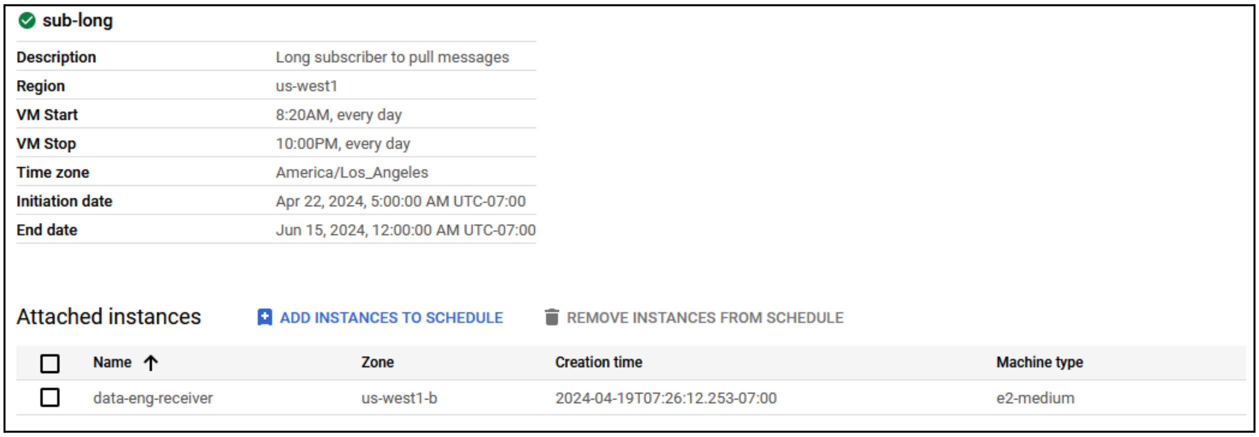


✓ daily-collection

| | |
|---|---|
| Description | collect data eng data daily |
| Region | us-west1 |
| VM Start | 8:00AM, every day |
| VM Stop | 12:00PM, every day |
| Time zone | America/Los_Angeles |
| Initiation date | Apr 17, 2024, 12:00:00 AM UTC-07:00 |
| End date | Jun 15, 2024, 12:00:00 AM UTC-07:00 |

Attached instances

| □ | Name ↑ | Zone | Creation time | Machine type |
|---|---|---|---|---|
| □ | data-eng-vm | us-west1-b | 2024-04-03T11:56:38.085-07:00 | e2-medium |



✓ sub-long

| | |
|---|---|
| Description | Long subscriber to pull messages |
| Region | us-west1 |
| VM Start | 8:20AM, every day |
| VM Stop | 10:00PM, every day |
| Time zone | America/Los_Angeles |
| Initiation date | Apr 22, 2024, 5:00:00 AM UTC-07:00 |
| End date | Jun 15, 2024, 12:00:00 AM UTC-07:00 |

Attached instances

| □ | Name ↑ | Zone | Creation time | Machine type |
|---|---|---|---|---|
| □ | data-eng-receiver | us-west1-b | 2024-04-19T07:26:12.253-07:00 | e2-medium |