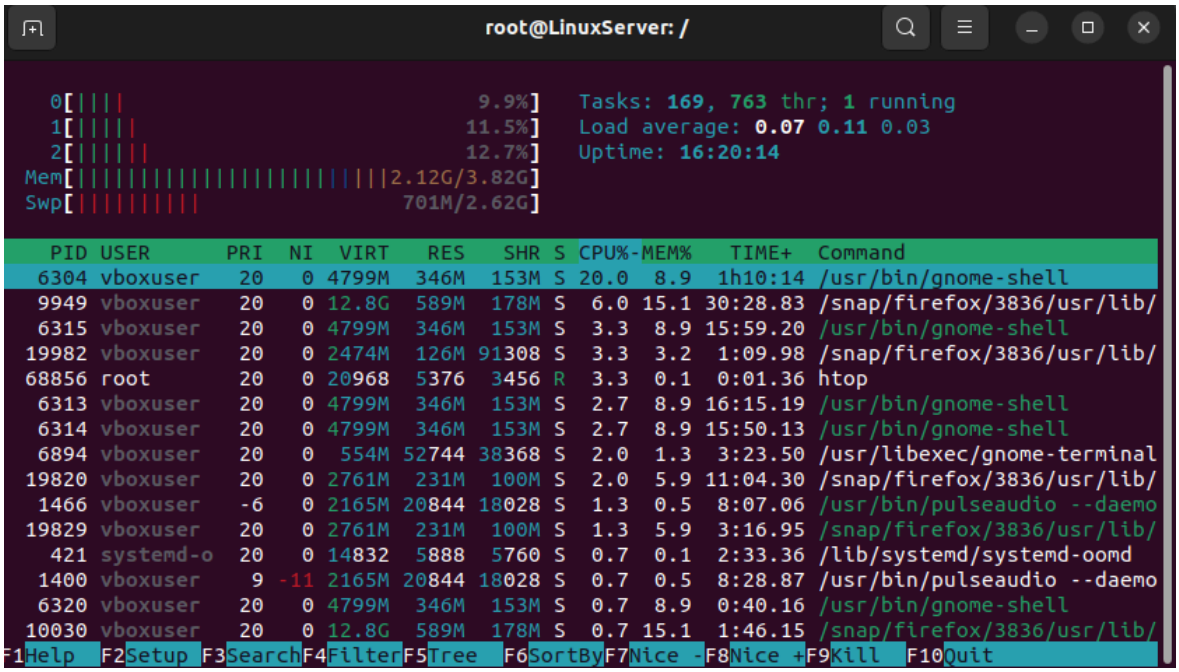
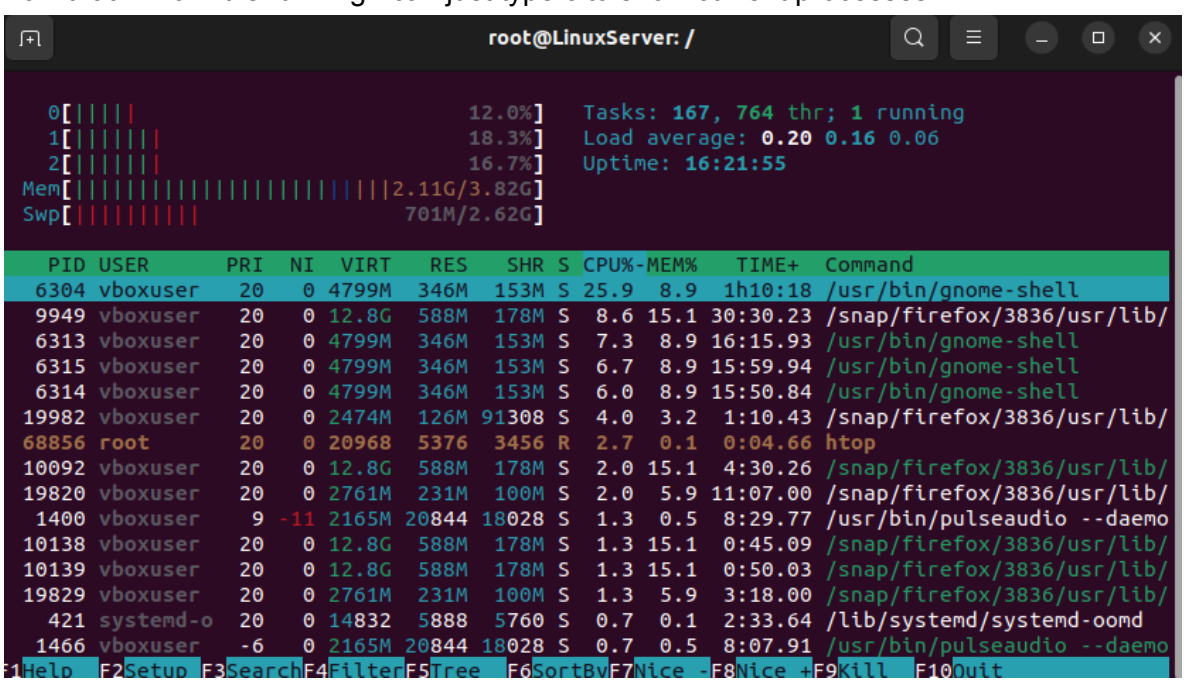
Part 2 Monitoring Own System:

First I will install htop to run basic commands to get information on basics system performances etc.

- I run sudo apt install htop
- Then i just run htop



- Now that I know it is running I can just type c to show current processes:



Here we can see several things about our host metric:

- First we can see the percentage used from each of our three cores (0 is 12%, 1 is 18.3%, 2 is 16.7%).
- We have 2.11 G of memory being used
- We see pour uptime being 16:21:55
- Here our load average is 0.20 for a minute, 0.16 for five, 0.06 for fifteen.
- Our swap space being used is 701 M.

- Here I can see several things, I can view the Uptime on my machine, the load averages (the first is a minute, the second is 5 minutes, the third is 15 min) and the % of CPU time a process is using (0% means it is not using the CPU at all, 100% means using a full CPU, 200% using two full CPUs and so on until 400% as our maximum CPUs are 4).

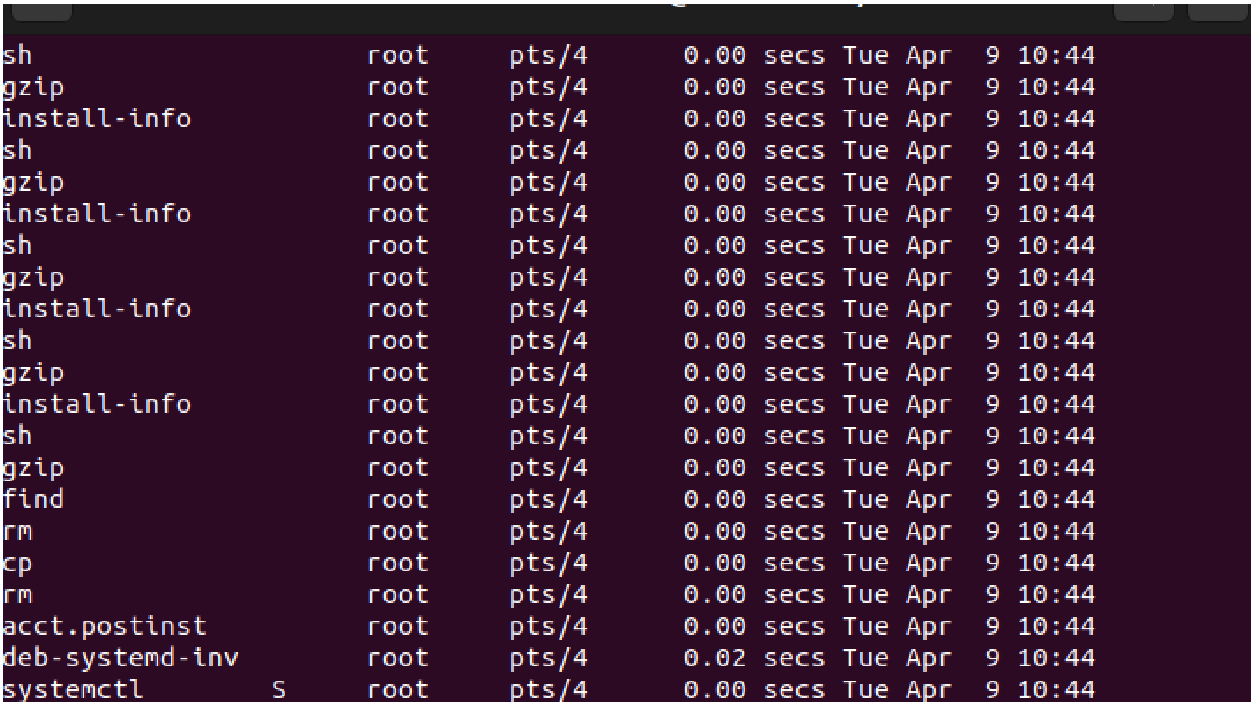
Next we will monitor user activity using ACCT:

- First we need to install it running sudo apt install acct install.
- Then we enable it running sudo systemctl enable acct.
- And start it sudo systemctl start acct.
- Then we can run sudo systemctl status acct to check that acct is active, we should have something like this:

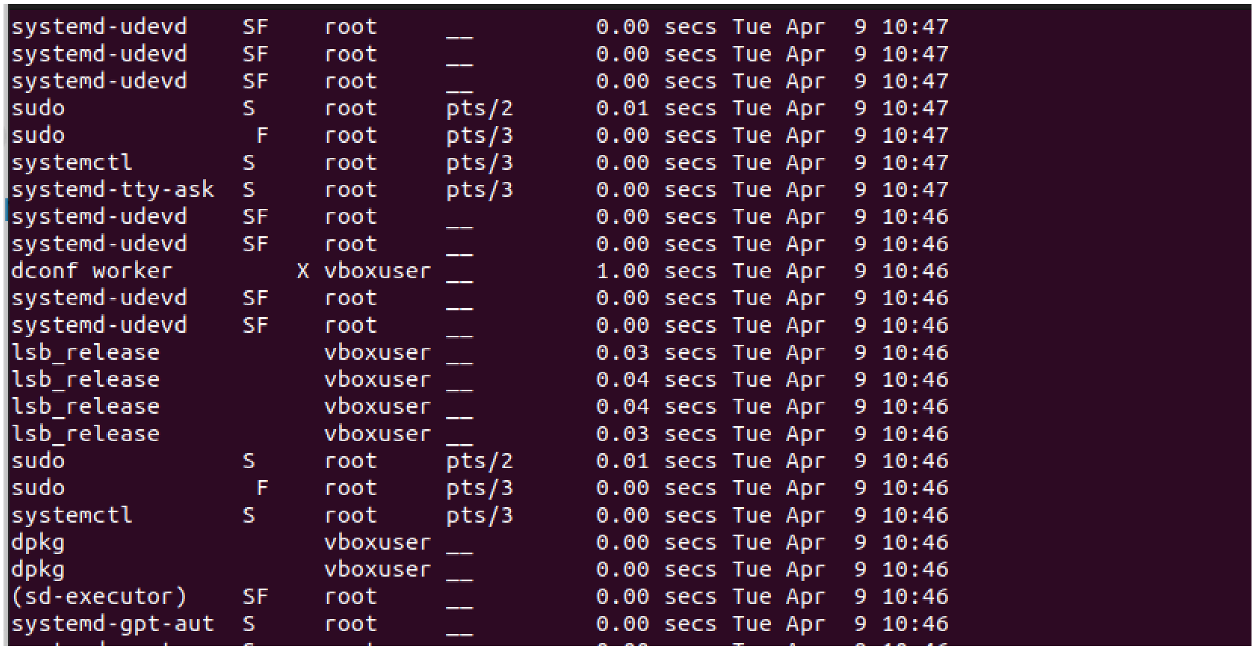
```
Created symlink /etc/systemd/system/multi-user.target.wants/acct.service -> /lib/systemd/sys
tem/acct.service.
Processing triggers for install-info (6.8-4build1) ...
Processing triggers for man-db (2.10.2-1) ...
root@LinuxServer:/# sudo systemctl enable psacct
Failed to enable unit: Unit file psacct.service does not exist.
root@LinuxServer:/# sudo systemctl enable acct
Synchronizing state of acct.service with SysV service script with /lib/systemd/systemd-sysv-
install.
Executing: /lib/systemd/systemd-sysv-install enable acct
root@LinuxServer:/# sudo systemctl start acct
root@LinuxServer:/# sudo systemctl status acct
* acct.service - Kernel process accounting
     Loaded: loaded (/lib/systemd/system/acct.service; enabled; vendor preset: enabled)
     Active: active (exited) since Tue 2024-04-09 10:44:59 CEST; 3min 40s ago
       Docs: man:accton(8)
   Main PID: 69957 (code=exited, status=0/SUCCESS)
        CPU: 2ms

apr 09 10:44:59 LinuxServer systemd[1]: Starting Kernel process accounting...
apr 09 10:44:59 LinuxServer accton[69957]: Turning on process accounting, file set to '/var>
apr 09 10:44:59 LinuxServer systemd[1]: Finished Kernel process accounting.
```

Here we see that our process is active!

- Then we can use the lastcomm command to view the logs and get a general view of the logs showing user activity mine will mostly show root:



```
sh                     root     pts/4      0.00 secs Tue Apr  9 10:44
gzip                   root     pts/4      0.00 secs Tue Apr  9 10:44
install-info           root     pts/4      0.00 secs Tue Apr  9 10:44
sh                     root     pts/4      0.00 secs Tue Apr  9 10:44
gzip                   root     pts/4      0.00 secs Tue Apr  9 10:44
install-info           root     pts/4      0.00 secs Tue Apr  9 10:44
sh                     root     pts/4      0.00 secs Tue Apr  9 10:44
gzip                   root     pts/4      0.00 secs Tue Apr  9 10:44
install-info           root     pts/4      0.00 secs Tue Apr  9 10:44
sh                     root     pts/4      0.00 secs Tue Apr  9 10:44
gzip                   root     pts/4      0.00 secs Tue Apr  9 10:44
install-info           root     pts/4      0.00 secs Tue Apr  9 10:44
sh                     root     pts/4      0.00 secs Tue Apr  9 10:44
gzip                   root     pts/4      0.00 secs Tue Apr  9 10:44
find                   root     pts/4      0.00 secs Tue Apr  9 10:44
rm                     root     pts/4      0.00 secs Tue Apr  9 10:44
cp                     root     pts/4      0.00 secs Tue Apr  9 10:44
rm                     root     pts/4      0.00 secs Tue Apr  9 10:44
acct.postinst          root     pts/4      0.00 secs Tue Apr  9 10:44
deb-systemd-inv        root     pts/4      0.02 secs Tue Apr  9 10:44
systemctl         S    root     pts/4      0.00 secs Tue Apr  9 10:44
```



```
systemd-udevd    SF    root     __         0.00 secs Tue Apr  9 10:47
systemd-udevd    SF    root     __         0.00 secs Tue Apr  9 10:47
systemd-udevd    SF    root     __         0.00 secs Tue Apr  9 10:47
sudo             S     root     pts/2      0.01 secs Tue Apr  9 10:47
sudo              F    root     pts/3      0.00 secs Tue Apr  9 10:47
systemctl        S     root     pts/3      0.00 secs Tue Apr  9 10:47
systemd-tty-ask  S     root     pts/3      0.00 secs Tue Apr  9 10:47
systemd-udevd    SF    root     __         0.00 secs Tue Apr  9 10:46
systemd-udevd    SF    root     __         0.00 secs Tue Apr  9 10:46
dconf worker         X vboxuser __         1.00 secs Tue Apr  9 10:46
systemd-udevd    SF    root     __         0.00 secs Tue Apr  9 10:46
systemd-udevd    SF    root     __         0.00 secs Tue Apr  9 10:46
lsb_release            vboxuser __         0.03 secs Tue Apr  9 10:46
lsb_release            vboxuser __         0.04 secs Tue Apr  9 10:46
lsb_release            vboxuser __         0.04 secs Tue Apr  9 10:46
lsb_release            vboxuser __         0.03 secs Tue Apr  9 10:46
sudo             S     root     pts/2      0.01 secs Tue Apr  9 10:46
sudo              F    root     pts/3      0.00 secs Tue Apr  9 10:46
systemctl        S     root     pts/3      0.00 secs Tue Apr  9 10:46
dpkg                   vboxuser __         0.00 secs Tue Apr  9 10:46
dpkg                   vboxuser __         0.00 secs Tue Apr  9 10:46
(sd-executor)    SF    root     __         0.00 secs Tue Apr  9 10:46
systemd-gpt-aut  S     root     __         0.00 secs Tue Apr  9 10:46
```

Here we can see two users' activity vboxuser and root, alongside what processes they are running etc.

- We can also use the w command to see the users logged in, the uptime and other info.

```
oot@LinuxServer:/# w
11:05:00 up 16:55,  3 users,  load average: 0,02, 0,13, 0,15
SER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
oxuser tty2     tty2              wo11    4days  0.02s  0.02s /usr/libexec/gnome-session-
oot     pts/1    -                do18    1.00s  0.43s  0.43s ssh localhost
oot     pts/2    127.0.0.1        do18    1.00s  2.27s  0.01s w
oot@LinuxServer:/#
```

- JCPU shows us the joint CPU runtime whichis the total amount of CPU time used by all processes attached to the user's session.
- PCPU: Displays the percentage of CPU time used by the user's processes since the last update of w.
- What: Shows the current command or activity being performed by the user. This typically includes the name of the command or program the user is running.
- User: Lists the username of each logged-in user.
- TTY: Indicates the terminal (or pseudo-terminal) associated with each user session.
- From: Displays the remote hostname or IP address from which a user is logged in, if applicable. If a user is logged in locally, it usually shows -.
- Login Time: Shows the time when each user logged in.
- Idle Time: Indicates how long each user has been idle. If a user has been active recently, it shows `0:00`.
- In this example we can see we have 3 users and that our average load time for a minute is 0.02, for five it is 0.13 and finally for fifteen is 0.15

Now let's have a look at prometheus, I will mention later how it can be used to analyze different metrics mentioned in part one:

- First, we want to install prometheus to take a look at my application metrics
- To do so we go on the prometheus website and select the correct file to add to our system.

2.51.1 / 2024-03-27 Release notes

File name	OS	Arch	Size	SHA256 Checksum
prometheus-2.51.1.darwin-amd64.tar.gz	darwin	amd64	97.42 MiB	f8046dd097538ba71c33fa6cfd83d8fb4bf5f58c901e58ec0c66288ec95db4bd
prometheus-2.51.1.linux-amd64.tar.gz	linux	amd64	96.98 MiB	1f933ea7515e3a6e60374ee0bfdb62bc4701c7b12c1dbafe1865c327c6e0e7d2
prometheus-2.51.1.windows-amd64.zip	windows	amd64	99.13 MiB	15cc0d176ccc5149ffaa895440e325db7a4ceb3c4804913c635b1215dff31e8e

2.45.4 / 2024-03-18 LTS Release notes

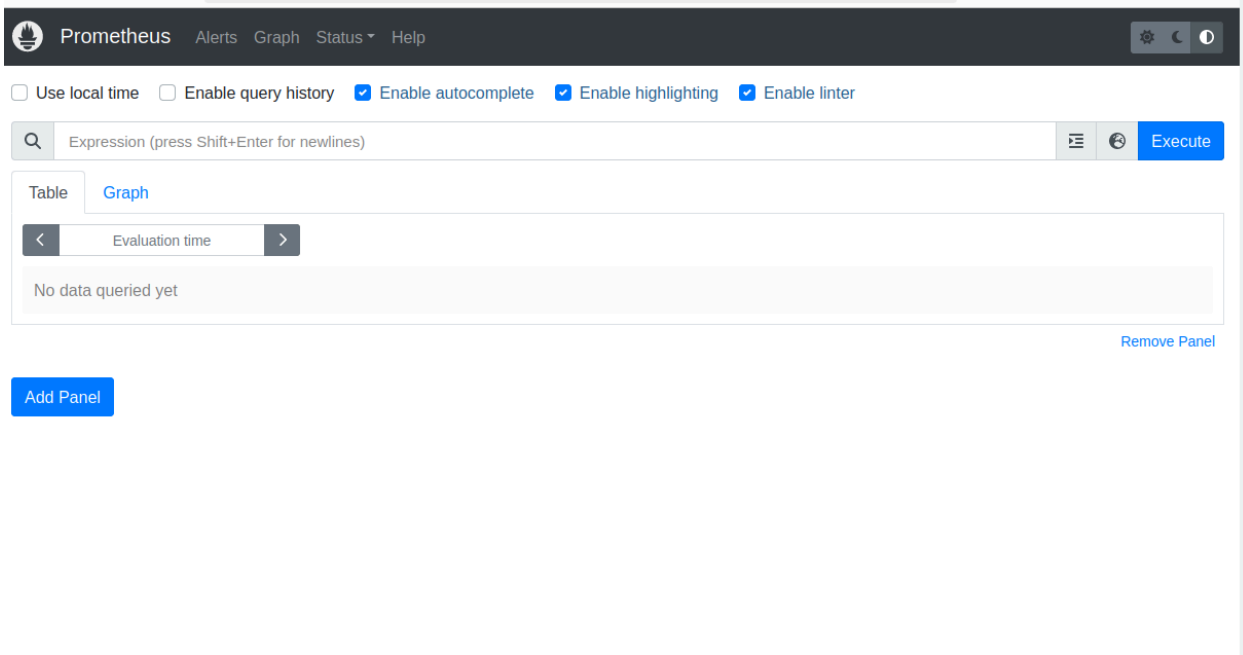
File name	OS	Arch	Size	SHA256 Checksum

- Then we go into the directory where the file was downloaded and unzip it running tar xvfz /prometheus/prometheus-2.30.3.linux-amd64/

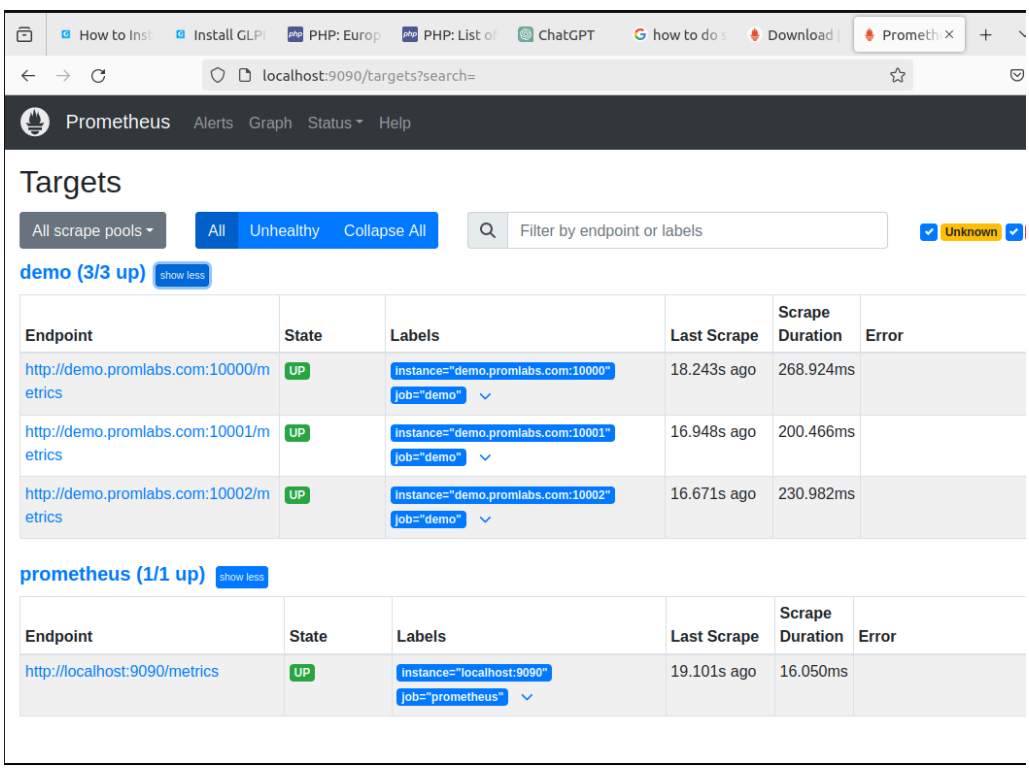
- The go into the folder cd prometheus-2.30.3.linux-amd64/
- Change the config file by running sudo nano prometheus.yml according to what we need, here I will put the demo recommended config given by official documentation:

```
global:
  scrape_interval: 5s

scrape_configs:
  - job_name: "prometheus"
    static_configs:
      - targets:
          - localhost:9090

  - job_name: "demo"
    static_configs:
      - targets:
          - demo.promlabs.com:10000
          - demo.promlabs.com:10001
          - demo.promlabs.com:10002
```

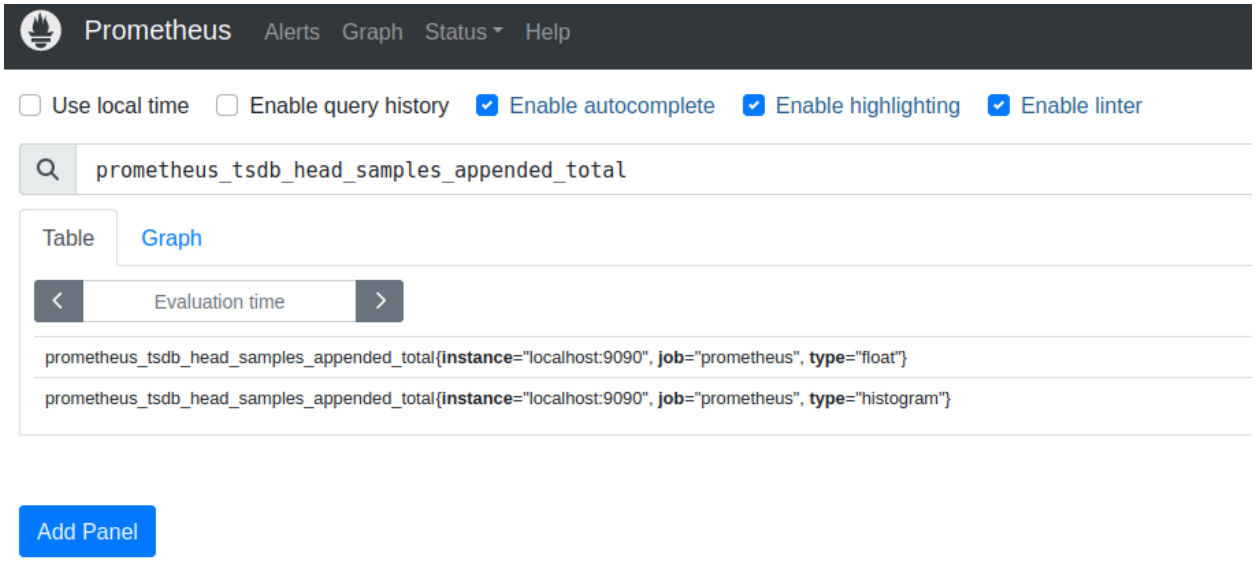
- And now run ./prometheus to start the server
- Now go to you browser of choice and go to localost:9090



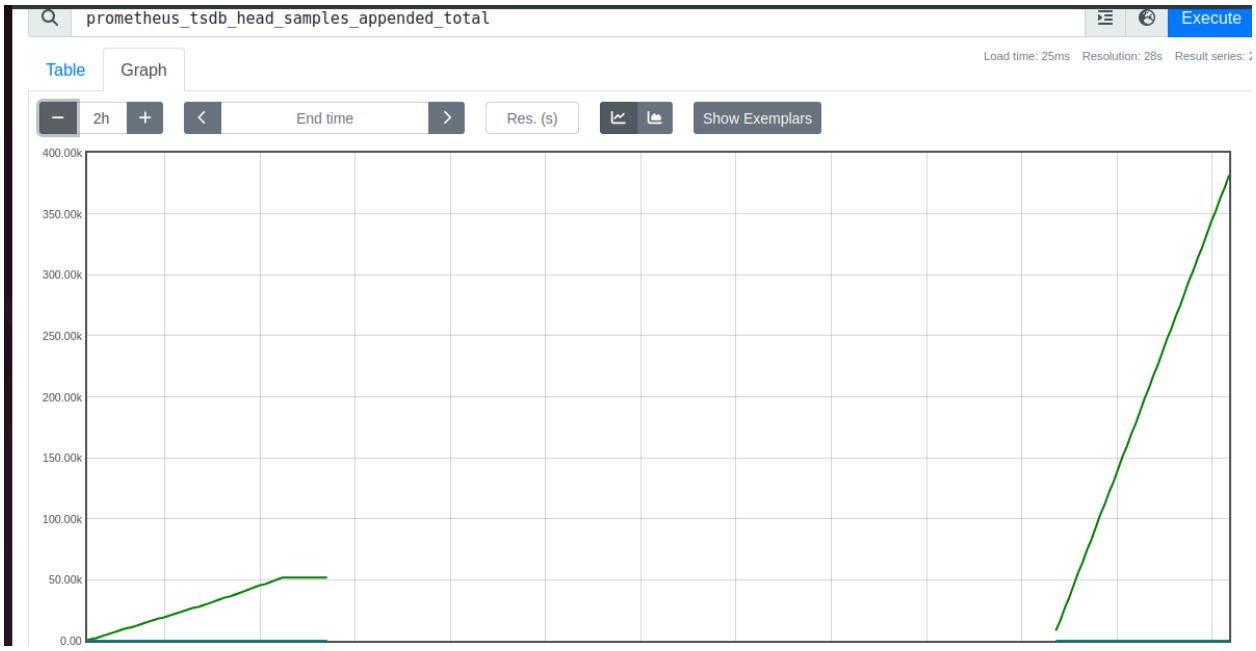
- Here we can see the prometheus UI. We can check under status then targets to see if they are up:



- All of them are up. Now we can focus on trying to look at a metric and the data Prometheus scraped on it:



- Here I entered the prometheus_tsdb_head_amples_appended_total (total number of samples that have been appended to the head of the time series database)
- In Prometheus, the "head" of the TSDB refers to the most recent data that has been ingested and stored in memory or the most actively queried portion of the time series data. This typically includes the most recent data points.
- Each time a new data point (sample) is ingested into Prometheus and stored at the most recent part of the TSDB (the head), this metric value increases.
- Monitoring this metric can provide insights into the rate at which new data is being collected and added to Prometheus. It's useful for understanding the data ingestion rate and the activity level of the monitoring system.



- Here we can see how the data has changed over the last 2 hours going into the graph section. Showing how we ccan keep track of different application metrics and monitor them using prometheus.
- This can be used from server metrics to External dependencies metrics to Application metrics. We can even set it up to scrape particular metrics of interest etc.

Let's look at our network performance:

- First we need to intsall the iftop tool: sudo apt-get install iftop.
- Then we run it using sudo iftop

```
root@LinuxServer:/home/vboxuser/Downloads/prometheus-2.51.1.linux-amd64# sudo iftop
interface: enp0s3
IP address is: 10.0.2.15
MAC address is: 08:00:27:ab:66:c8
root@LinuxServer:/home/vboxuser/Downloads/prometheus-2.51.1.linux-amd64#
```

- So here we can see basic info about our network

```
                  12,5Kb            25,0Kb            37,5Kb            50,0Kb       62,5Kb

TX:             cum:      0B    peak:      0b              rates:      0b      0b      0b
RX:                       0B               0b                          0b      0b      0b
TOTAL:                    0B               0b                          0b      0b      0b
```

- And here the tool is listening enp0s3 here we do not have much going on but if that changes we will see it on here and be able to keep track of efficiency etc for our network.

Now lets take a look at events:

- Here we do not need a specific tool we can just look at our logs by running cat /var/log/syslog.

```
Apr  9 14:17:48 LinuxServer gnome-shell[6304]: Window manager warning: Overwriting existing
binding of keysym 32 with keysym 32 (keycode b).
Apr  9 14:17:48 LinuxServer gnome-shell[6304]: Window manager warning: Overwriting existing
binding of keysym 31 with keysym 31 (keycode a).
Apr  9 14:17:48 LinuxServer gnome-shell[6304]: Window manager warning: Overwriting existing
binding of keysym 33 with keysym 33 (keycode c).
Apr  9 14:17:48 LinuxServer gnome-shell[6304]: Window manager warning: Overwriting existing
binding of keysym 34 with keysym 34 (keycode d).
Apr  9 14:17:48 LinuxServer gnome-shell[6304]: Window manager warning: Overwriting existing
binding of keysym 36 with keysym 36 (keycode f).
Apr  9 14:17:48 LinuxServer gnome-shell[6304]: Window manager warning: Overwriting existing
binding of keysym 37 with keysym 37 (keycode 10).
Apr  9 14:17:48 LinuxServer gnome-shell[6304]: Window manager warning: Overwriting existing
binding of keysym 38 with keysym 38 (keycode 11).
Apr  9 14:17:48 LinuxServer gnome-shell[6304]: Window manager warning: Overwriting existing
binding of keysym 39 with keysym 39 (keycode 12).
Apr  9 14:17:48 LinuxServer dbus-daemon[5956]: [session uid=1000 pid=5956] Activating servic
e name='org.gnome.ArchiveManager1' requested by ':1.263' (uid=1000 pid=72480 comm="gjs /usr/
share/gnome-shell/extensions/ding@rasters" label="unconfined")
Apr  9 14:17:49 LinuxServer dbus-daemon[5956]: [session uid=1000 pid=5956] Successfully acti
vated service 'org.gnome.ArchiveManager1'
Apr  9 14:17:49 LinuxServer gnome-shell[6304]: DING: Detected async api for thumbnails
Apr  9 14:17:49 LinuxServer gnome-shell[6304]: DING: GNOME nautilus 42.6
```

- Here we can see my logs warning me of a recent even where existing binding is being overwritten

- We can also see that a service was activated successfully

And we pretty much covered everything covered everything in part one and tried it in this part! Hope this helps!