Term Project 3rd Submission

1. Select tools to be included into your codes or your system.
2. Screenshots of dashboard for various data types.
3. Screenshots of the use of analytics for the data obtained.
4. Screenshots of observability for overloads and crashes.

Group: Honey Lemon
Members:

1. 65011258 Arayan Tiwari
2. 65011269 Budthbundit Thaya-urai
3. 65011508 Ravinan Woraratyanon

Included tools

- **Telegraf:** Collects system metrics or data from MQTT and exposes them for Prometheus to scrape.
- **Node Exporter:** Monitor and collect the hardware and OS-level health of the system.
- **Prometheus:** Scrapes metrics from Telegraf and stores them for querying and alerting.
- **Grafana:** Visualizes the collected metrics from Prometheus.

docker-compose.yaml

```yaml
version: '3.8'

services:
 prometheus:
   image: prom/prometheus:latest
   container_name: prometheus
   ports:
     - "9090:9090"
   volumes:
     - ./prometheus/prometheus.yml:/etc/prometheus/prometheus.yml
   command:
     - '--config.file=/etc/prometheus/prometheus.yml'
   networks:
     - monitoring

 grafana:
   image: grafana/grafana:latest
   container_name: grafana
   ports:
     - "3000:3000"
   environment:
     - GF_SECURITY_ADMIN_USER=admin
     - GF_SECURITY_ADMIN_PASSWORD=admin
   volumes:
     - grafana-storage:/var/lib/grafana
   depends_on:
     - prometheus
   networks:
     - monitoring

 telegraf:
   image: telegraf:latest
   container_name: telegraf
   volumes:
     - ./telegraf/telegraf.conf:/etc/telegraf/telegraf.conf:ro
   ports:
     - "9273:9273"
   networks:
     - monitoring

 node_exporter:
   image: prom/node-exporter:latest
   container_name: node_exporter
   ports:
     - "9100:9100"
   networks:
     - monitoring
   restart: unless-stopped

networks:
 monitoring:
   driver: bridge

volumes:
 grafana-storage: {}
```

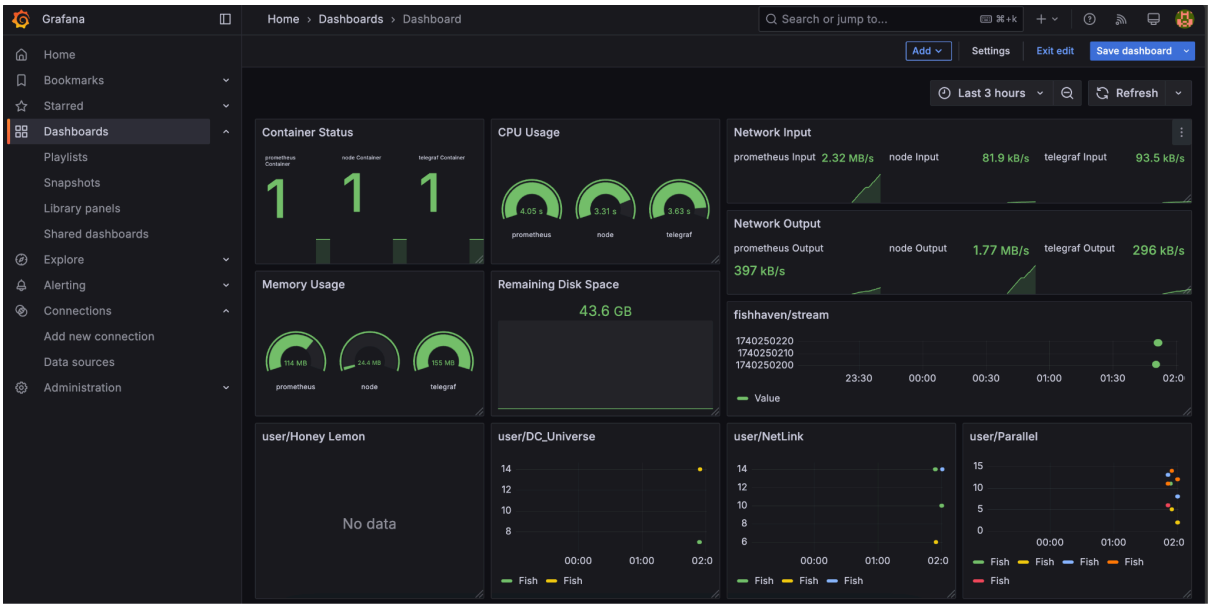
telegraf/telegraf.conf

```
[agent]
 interval = "10s"
 round_interval = true
 metric_batch_size = 1000
 metric_buffer_limit = 10000
 collection_jitter = "0s"
 flush_interval = "10s"
 flush_jitter = "0s"
 precision = ""
 debug = true  # Debug at agent level is supported

[[outputs.prometheus_client]]
 listen = ":9273"
 metric_version = 2

[[inputs.mqtt_consumer]]
 ## MQTT broker URLs
 servers = ["tcp://40.90.169.126:1883"]
  ## Credentials
 username = "dc24"
 password = "kmitl-dc24"
  ## Topics to subscribe to
 topics = [
   "fishhaven/stream",
   "user/Honey Lemon",
   "user/NetLink",
   "user/DC_Universe",
   "user/Parallel"
 ]
  ## Message handling
 data_format = "json"
  ## Add topic as a tag
 topic_tag = "topic"
  ## Enable QoS
 qos = 0
  ## Enable client ID
 client_id = "telegraf_mqtt_consumer"
  ## Fields to convert to tags
 tag_keys = ["type", "sender", "name", "group_name"]
  ## Persistent session to maintain subscription
 persistent_session = true
```

prometheus.yml

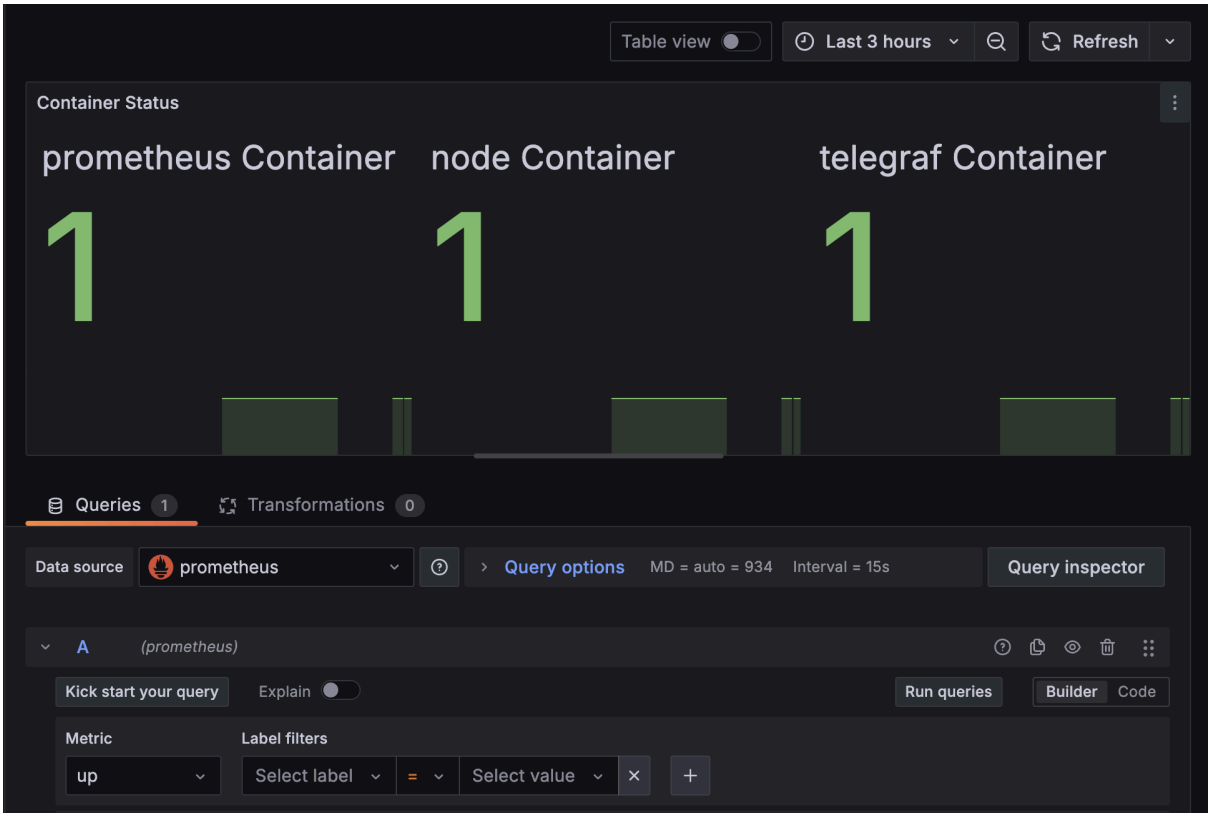
```yaml
global:
 scrape_interval: 15s
 evaluation_interval: 15s

scrape_configs:
 - job_name: 'telegraf'
   static_configs:
     - targets: ['telegraf:9273']

 - job_name: 'prometheus'
   static_configs:
     - targets: ['localhost:9090']

 - job_name: 'node'
   static_configs:
     - targets: ['node_exporter:9100']
```

Dashboard

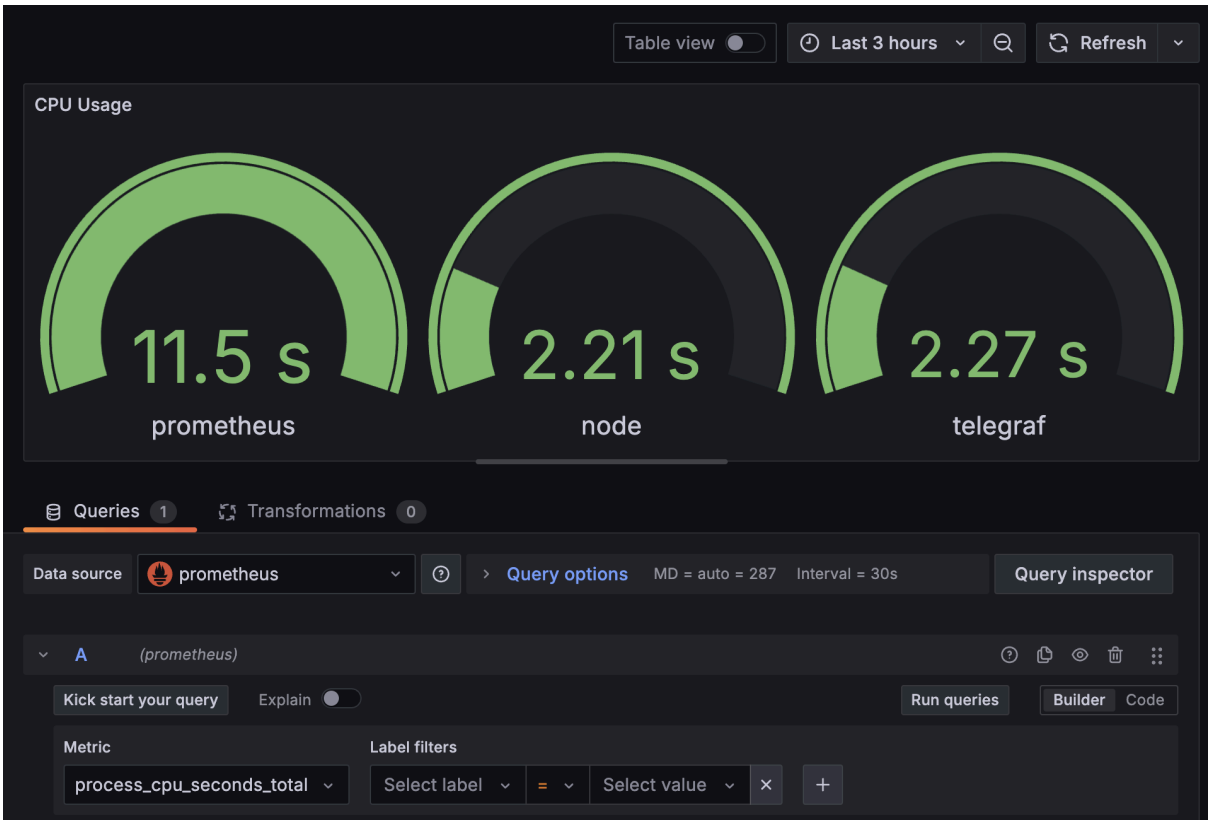

Grafana
Home
Bookmarks
Starred
Dashboards
Playlists
Snapshots
Library panels
Shared dashboards
Explore
Alerting
Connections
Add new connection
Data sources
Administration
Home › Dashboards › Dashboard
Search or jump to...
Add
Settings
Exit edit
Save dashboard
Last 3 hours
Refresh
Container Status
prometheus Container
node Container
telegraf Container
1
1
1
CPU Usage
4.05 s
3.31 s
3.63 s
prometheus
node
telegraf
Network Input
prometheus Input 2.32 MB/s
node Input 81.9 kB/s
telegraf Input 93.5 kB/s
Network Output
prometheus Output 397 kB/s
node Output 1.77 MB/s
telegraf Output 296 kB/s
Memory Usage
114 MB
24.4 MB
155 MB
prometheus
node
telegraf
Remaining Disk Space
43.6 GB
fishhaven/stream
1740250220
1740250210
1740250200
23:30
00:00
00:30
01:00
01:30
02:0
Value
user/Honey Lemon
No data
user/DC_Universe
14
12
10
8
00:00
01:00
02:0
Fish
Fish
user/NetLink
14
12
10
8
6
00:00
01:00
02:0
Fish
Fish
Fish
user/Parallel
15
10
5
0
00:00
01:00
02:0
Fish
Fish
Fish
Fish
Fish

Analytics

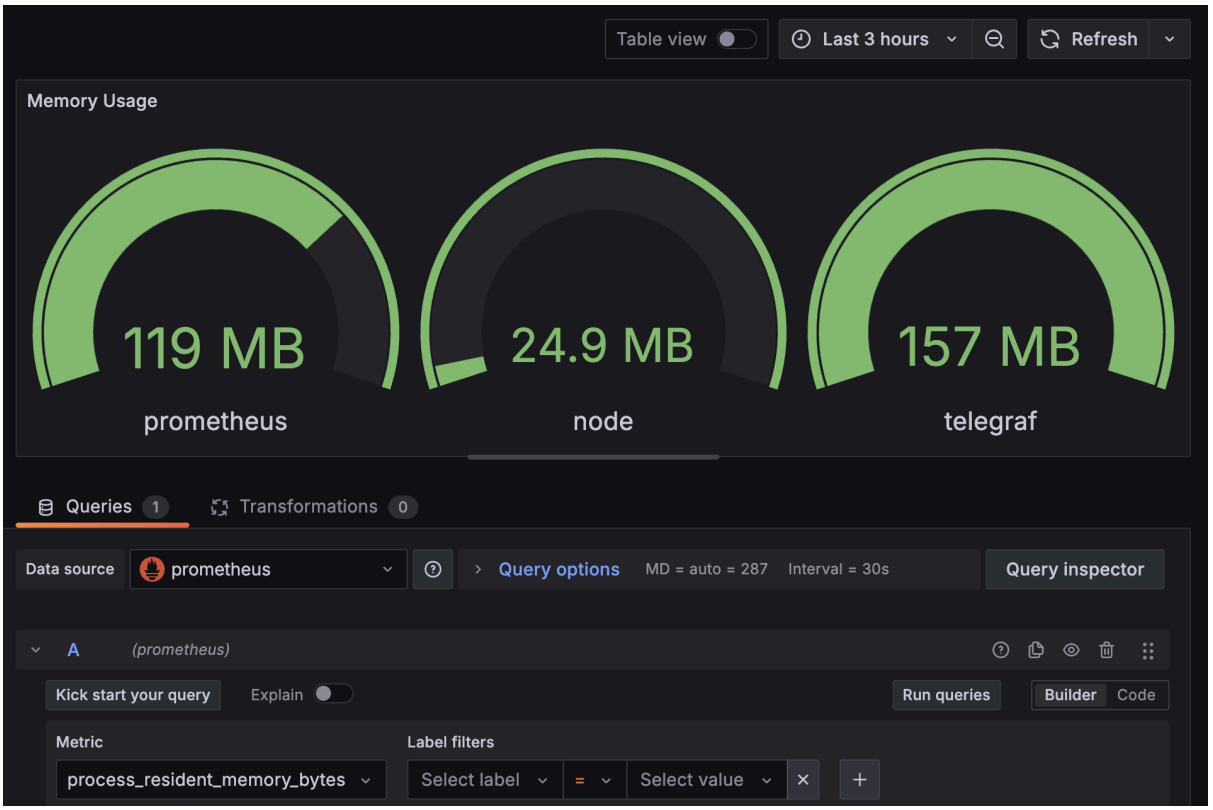
Container Status



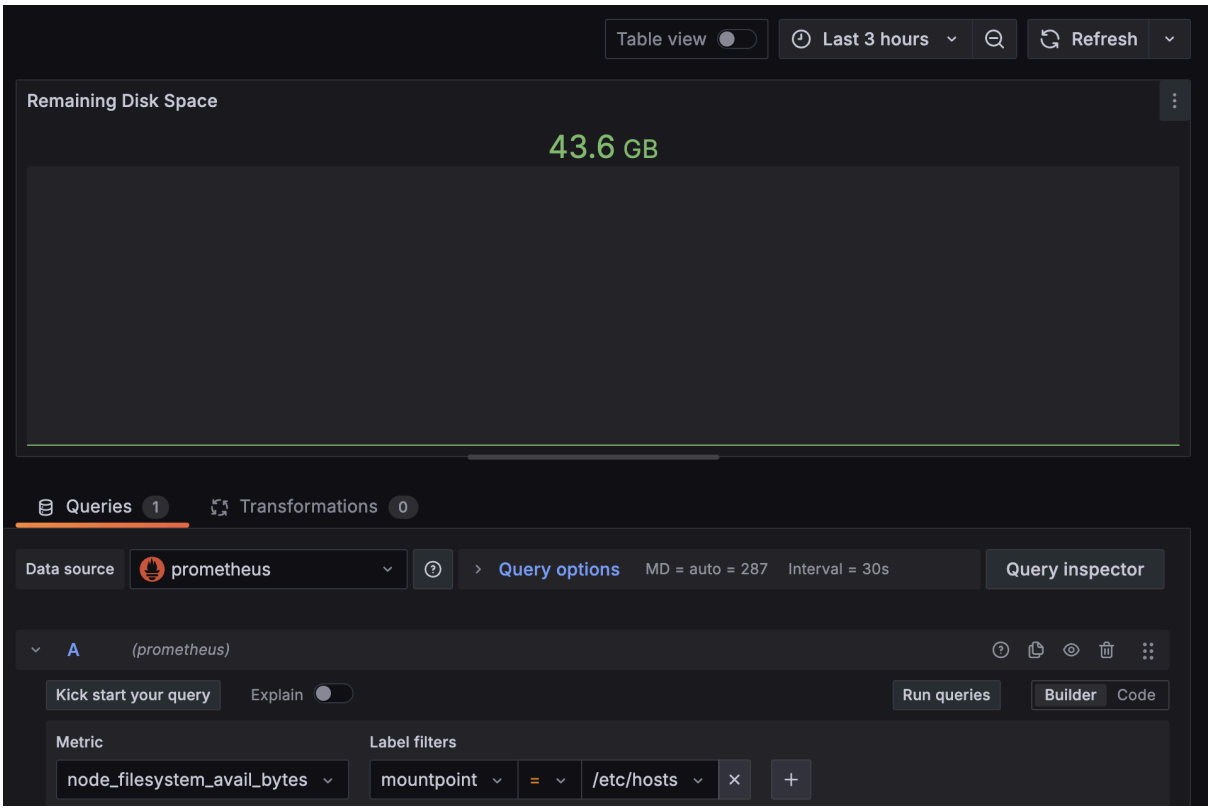
CPU usage for each tool



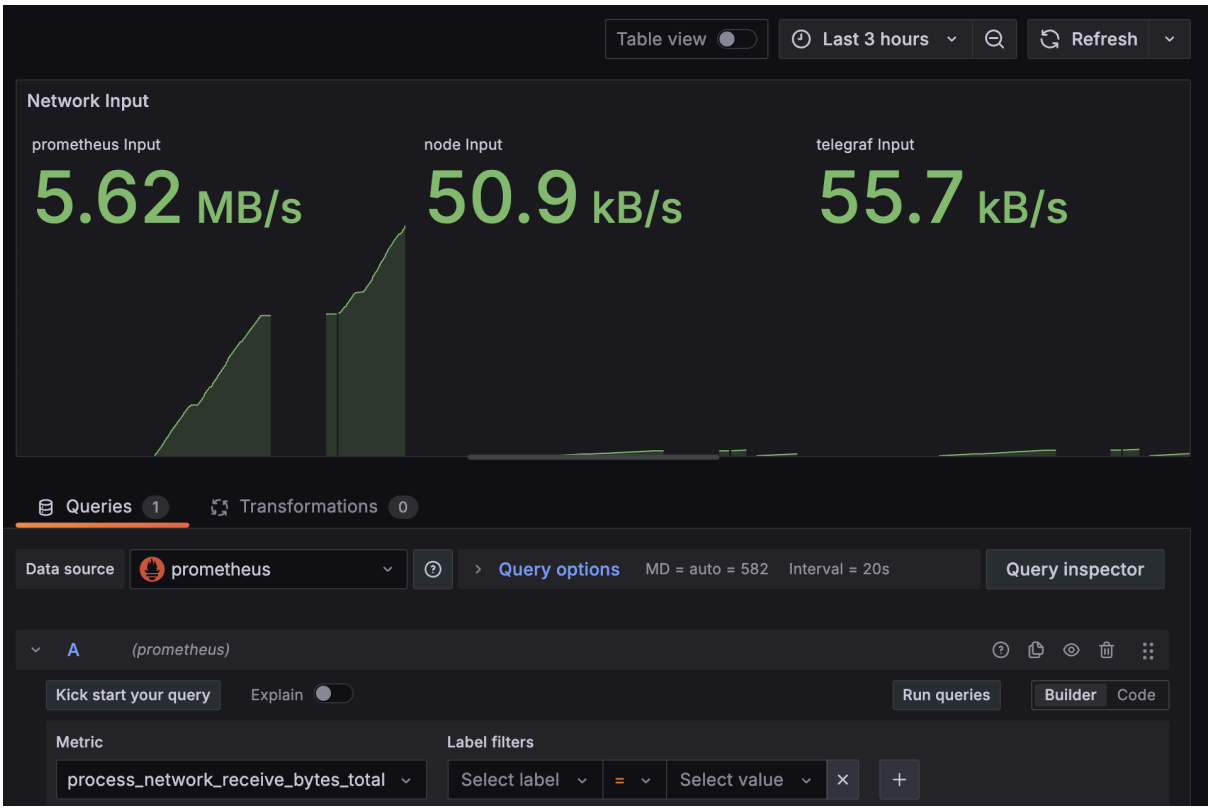
Memory consumption for each tool



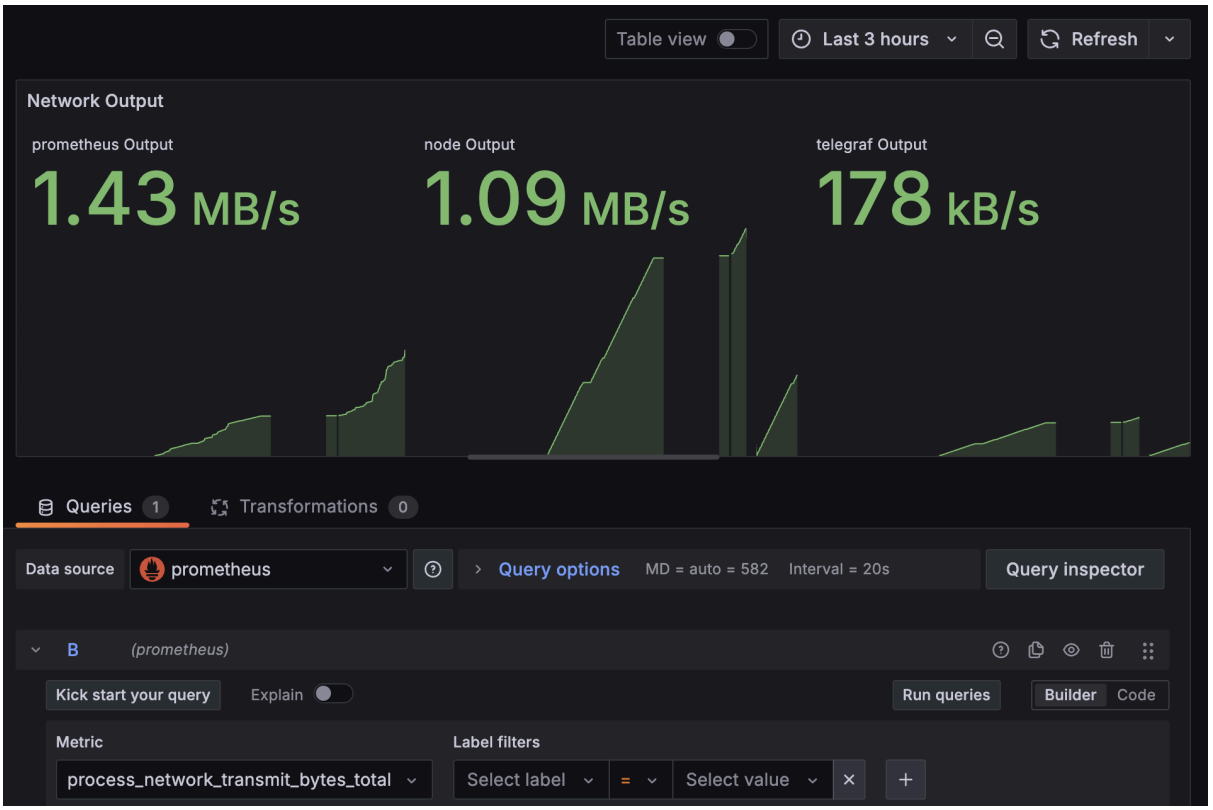
Available disk space



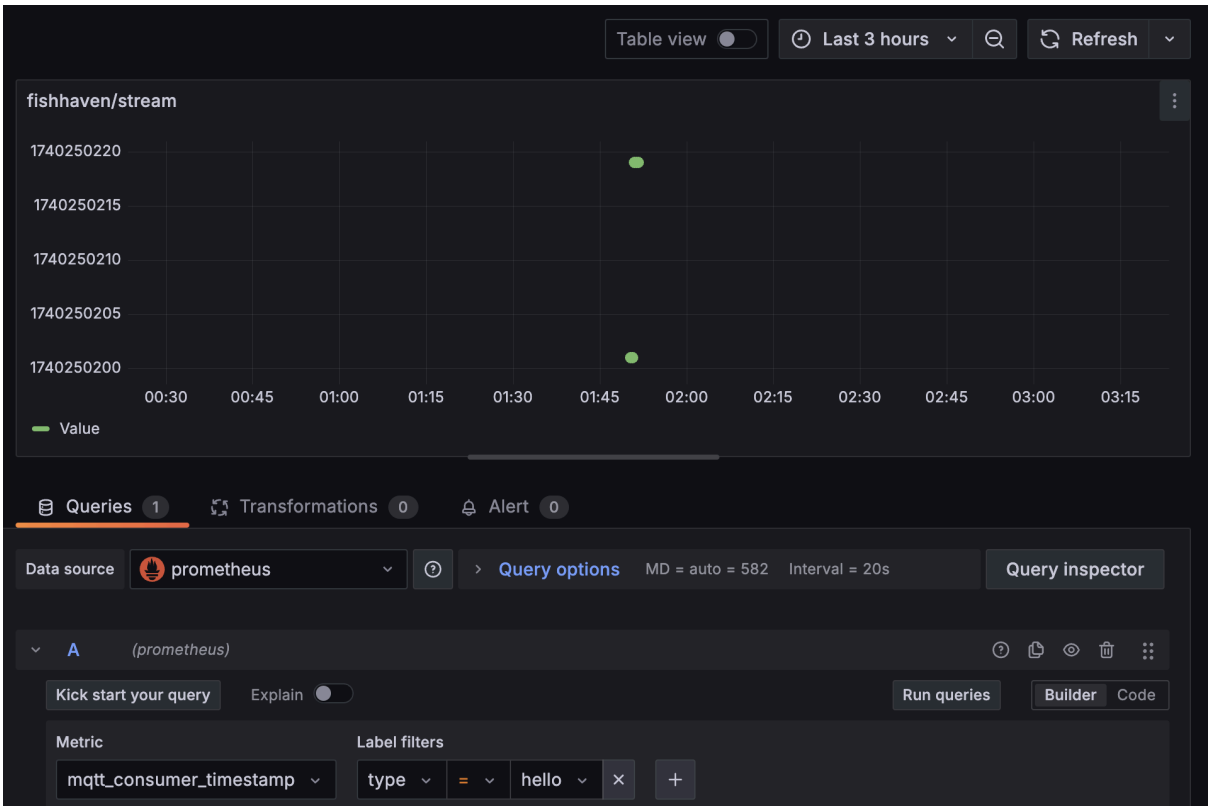
Network input rate for each tool



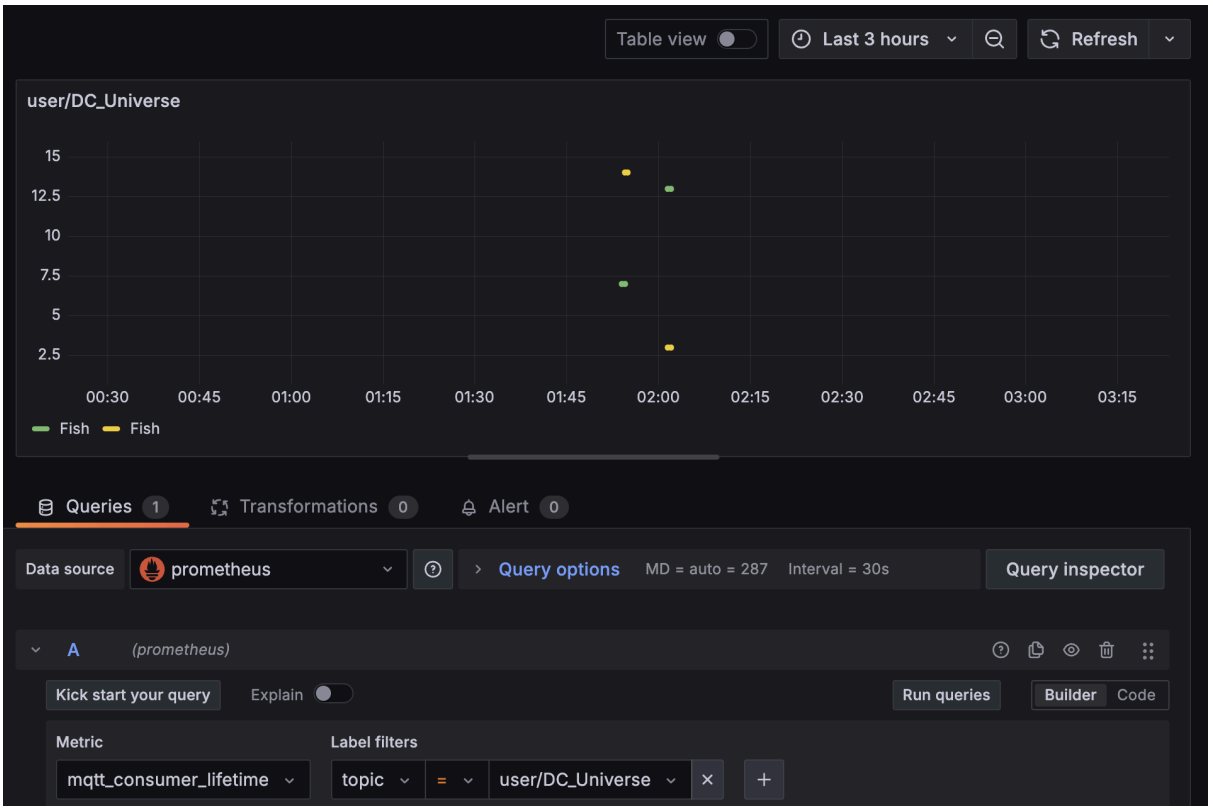
Network output rate for each tool



Monitor MQTT messages for the topic “fishhaven/stream”

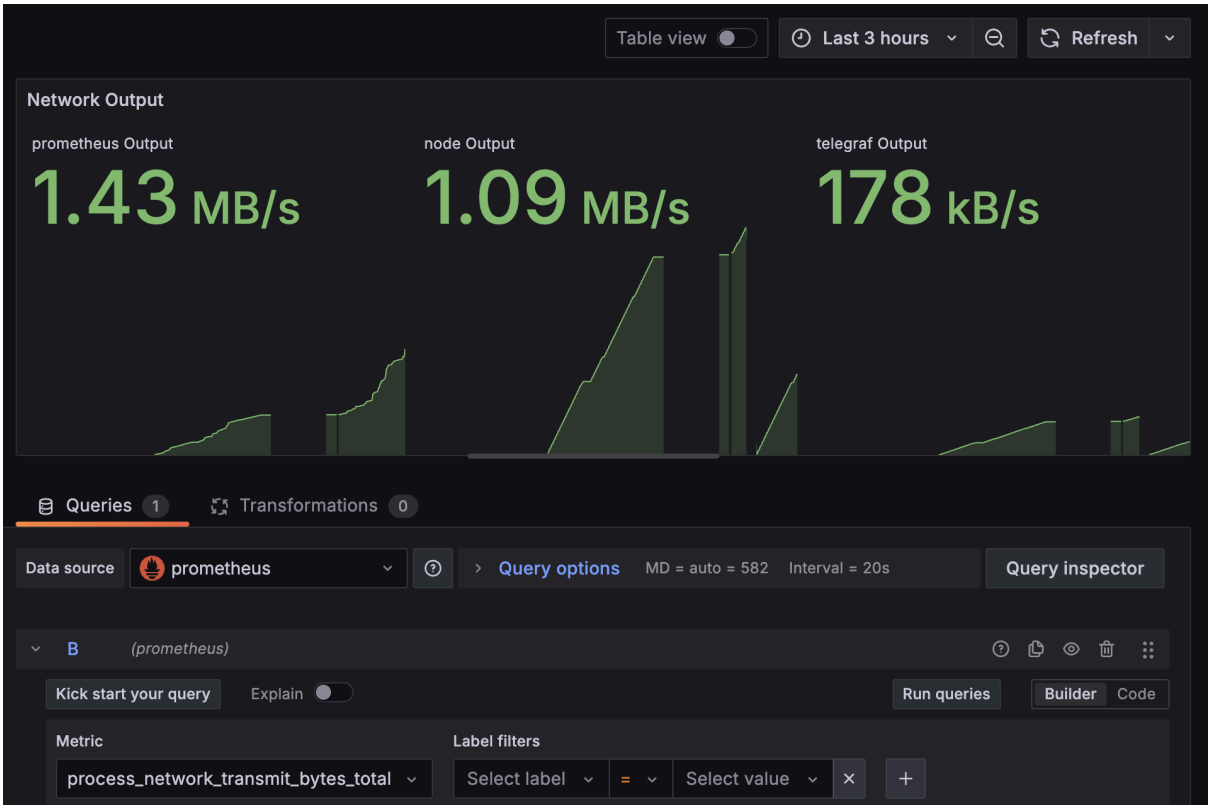


Monitor MQTT messages for the topic “user/{groupname}”

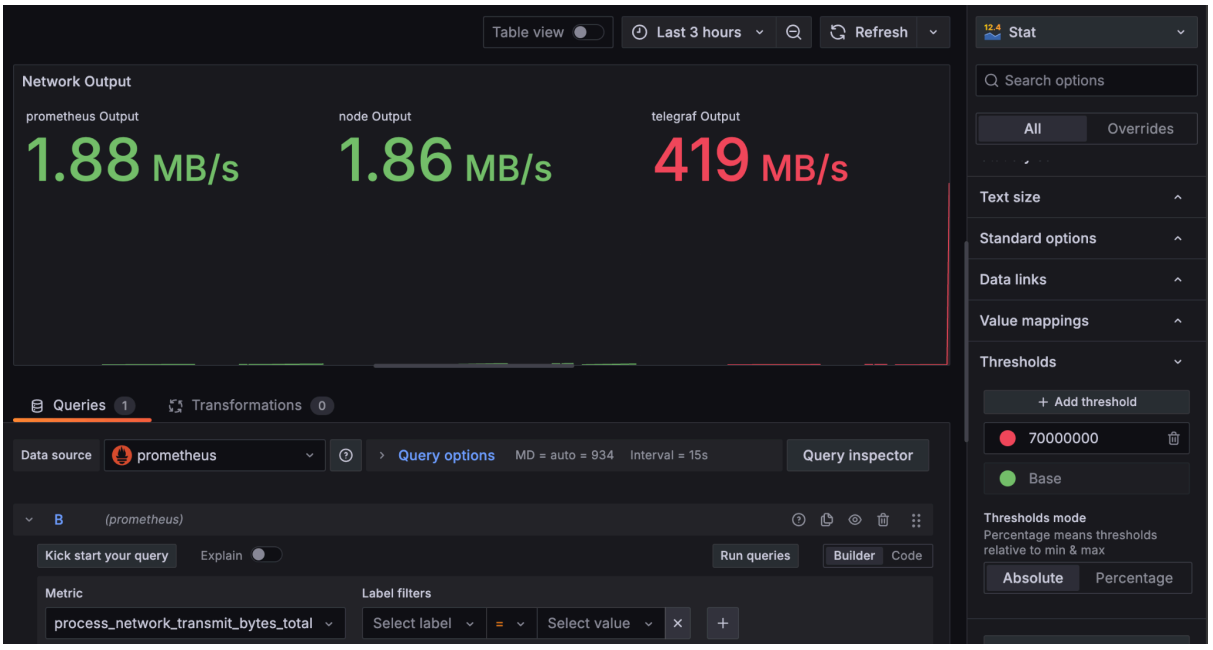


Overloads

Network output rate in normal condition

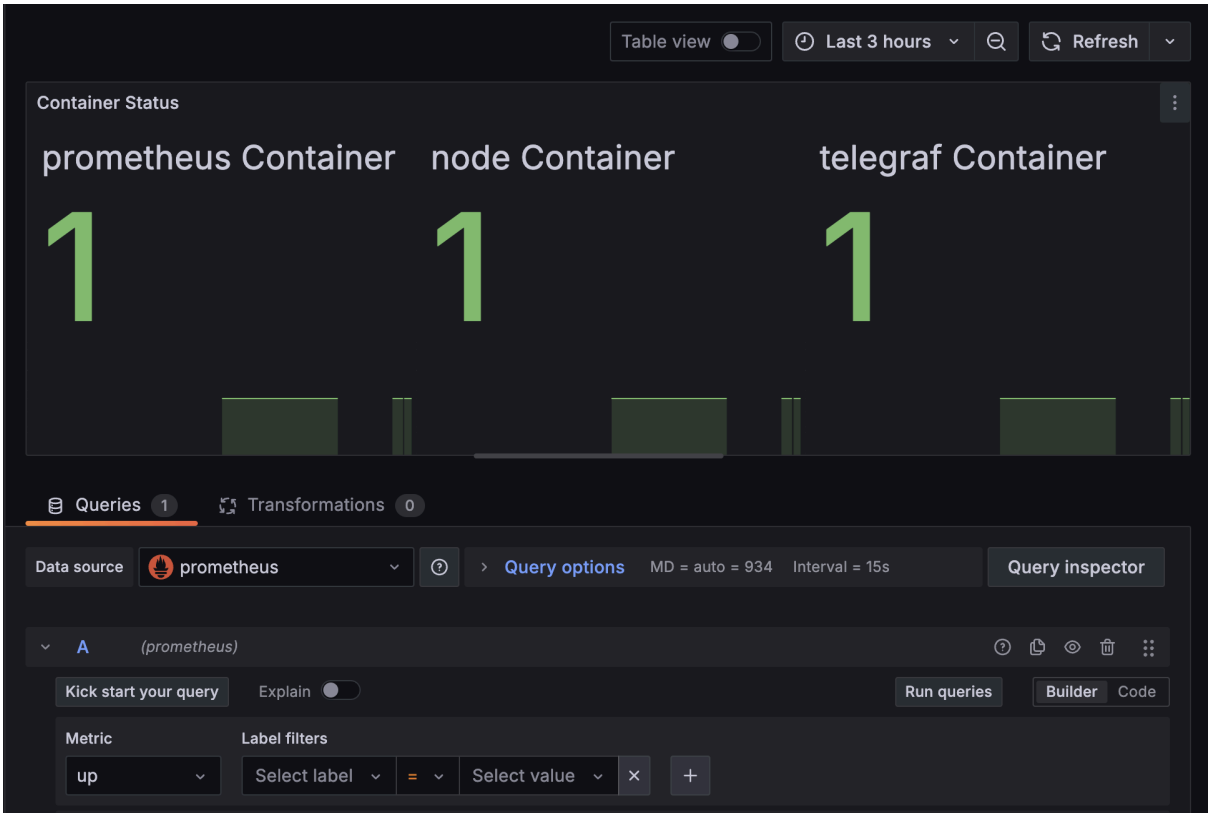


The network output rate in telegraf container is overloaded; set thresholds at 70MB.



Crashes

Container Status in normal condition



If container crashed (node and telegraf crashed)