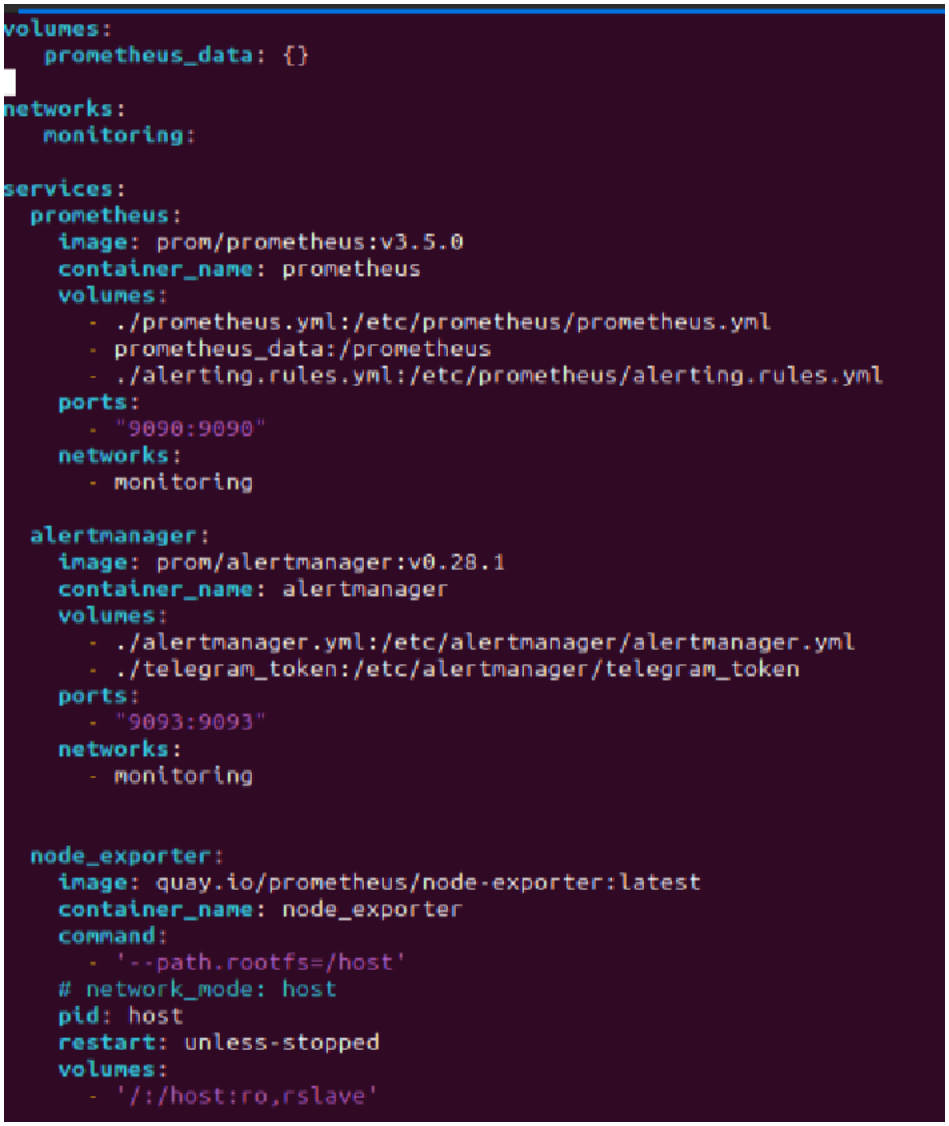
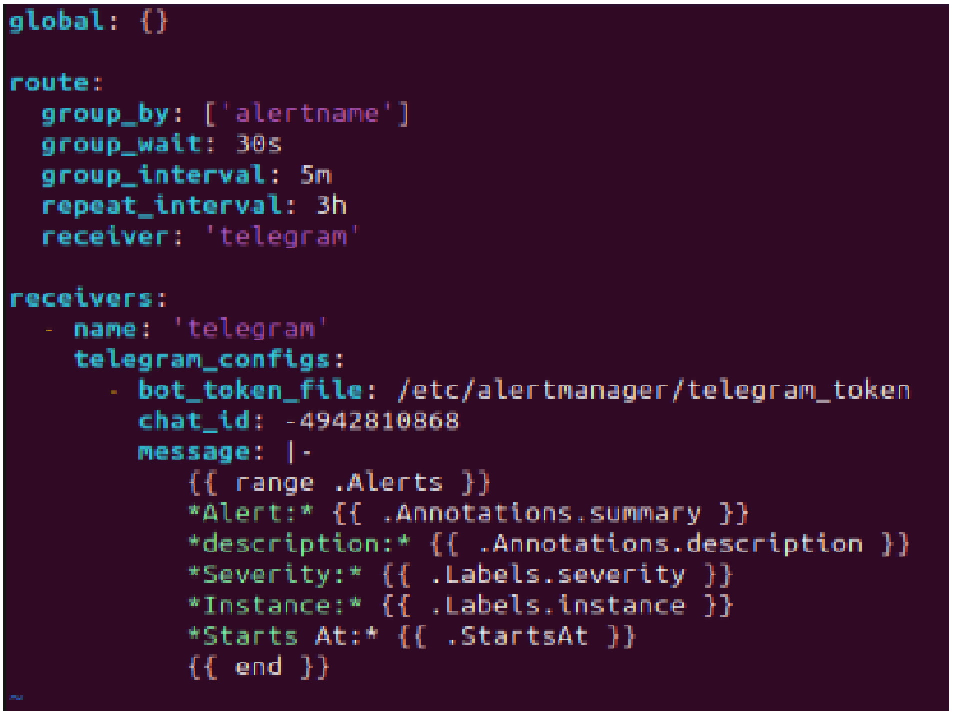
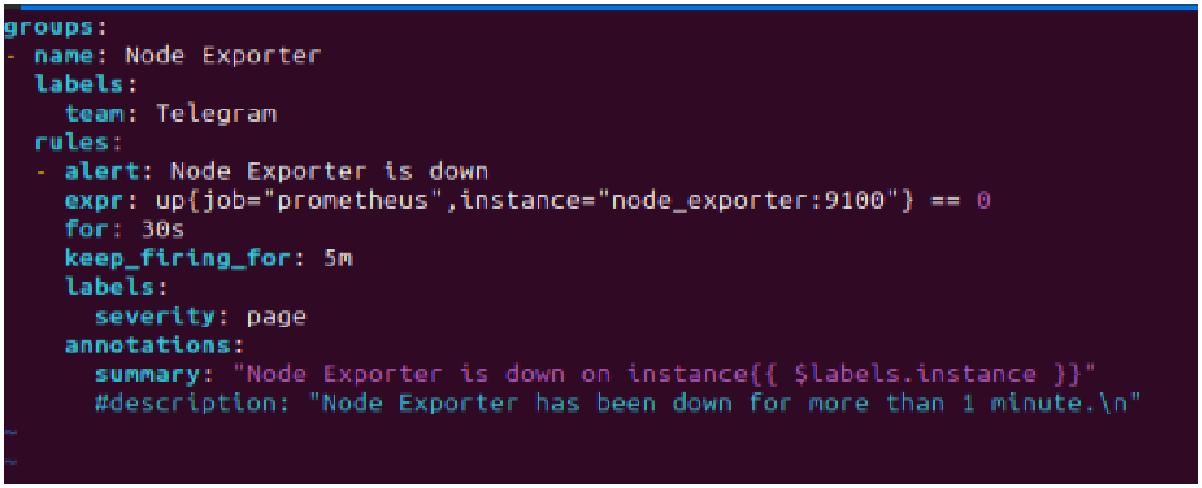
Task 1:



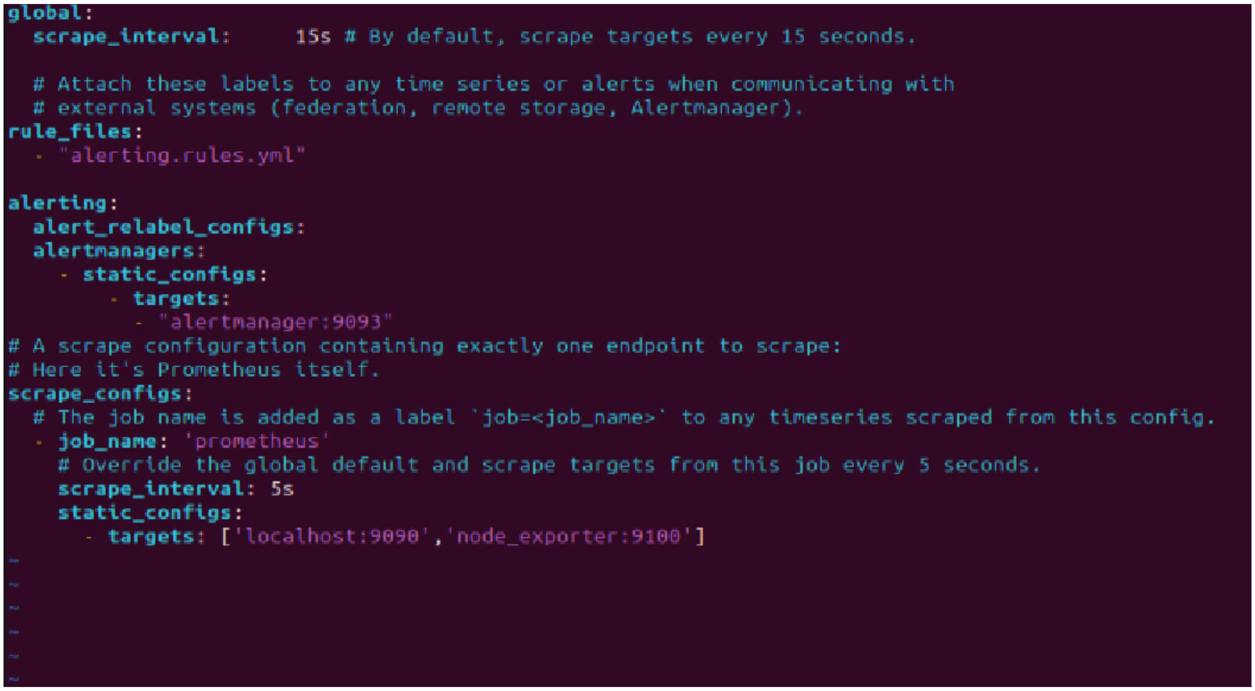
```
volumes:
   prometheus_data: {}

networks:
   monitoring:

services:
  prometheus:
    image: prom/prometheus:v3.5.0
    container_name: prometheus
    volumes:
      - ./prometheus.yml:/etc/prometheus/prometheus.yml
      - prometheus_data:/prometheus
      - ./alerting.rules.yml:/etc/prometheus/alerting.rules.yml
    ports:
      - "9090:9090"
    networks:
      - monitoring

  alertmanager:
    image: prom/alertmanager:v0.28.1
    container_name: alertmanager
    volumes:
      - ./alertmanager.yml:/etc/alertmanager/alertmanager.yml
      - ./telegram_token:/etc/alertmanager/telegram_token
    ports:
      - "9093:9093"
    networks:
      - monitoring


  node_exporter:
    image: quay.io/prometheus/node-exporter:latest
    container_name: node_exporter
    command:
      - '--path.rootfs=/host'
    # network_mode: host
    pid: host
    restart: unless-stopped
    volumes:
      - '/:/host:ro,rslave'
```



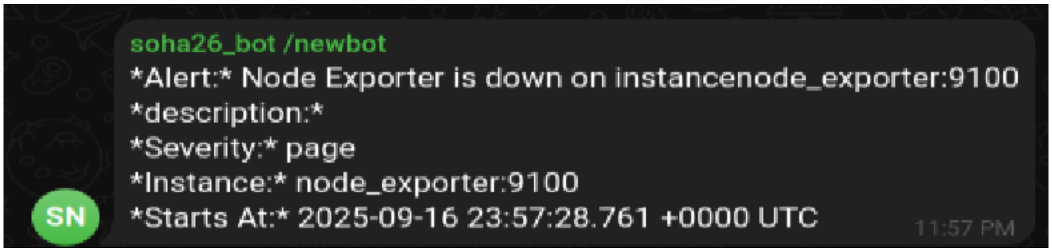
```
global: {}

route:
  group_by: ['alertname']
  group_wait: 30s
  group_interval: 5m
  repeat_interval: 3h
  receiver: 'telegram'

receivers:
  - name: 'telegram'
    telegram_configs:
      - bot_token_file: /etc/alertmanager/telegram_token
        chat_id: -4942810868
        message: |-
           {{ range .Alerts }}
           *Alert:* {{ .Annotations.summary }}
           *description:* {{ .Annotations.description }}
           *Severity:* {{ .Labels.severity }}
           *Instance:* {{ .Labels.instance }}
           *Starts At:* {{ .StartsAt }}
           {{ end }}
~
```



```
groups:
- name: Node Exporter
  labels:
    team: Telegram
  rules:
  - alert: Node Exporter is down
    expr: up{job="prometheus",instance="node_exporter:9100"} == 0
    for: 30s
    keep_firing_for: 5m
    labels:
      severity: page
    annotations:
      summary: "Node Exporter is down on instance{{ $labels.instance }}"
      #description: "Node Exporter has been down for more than 1 minute.\n"
~
~
```



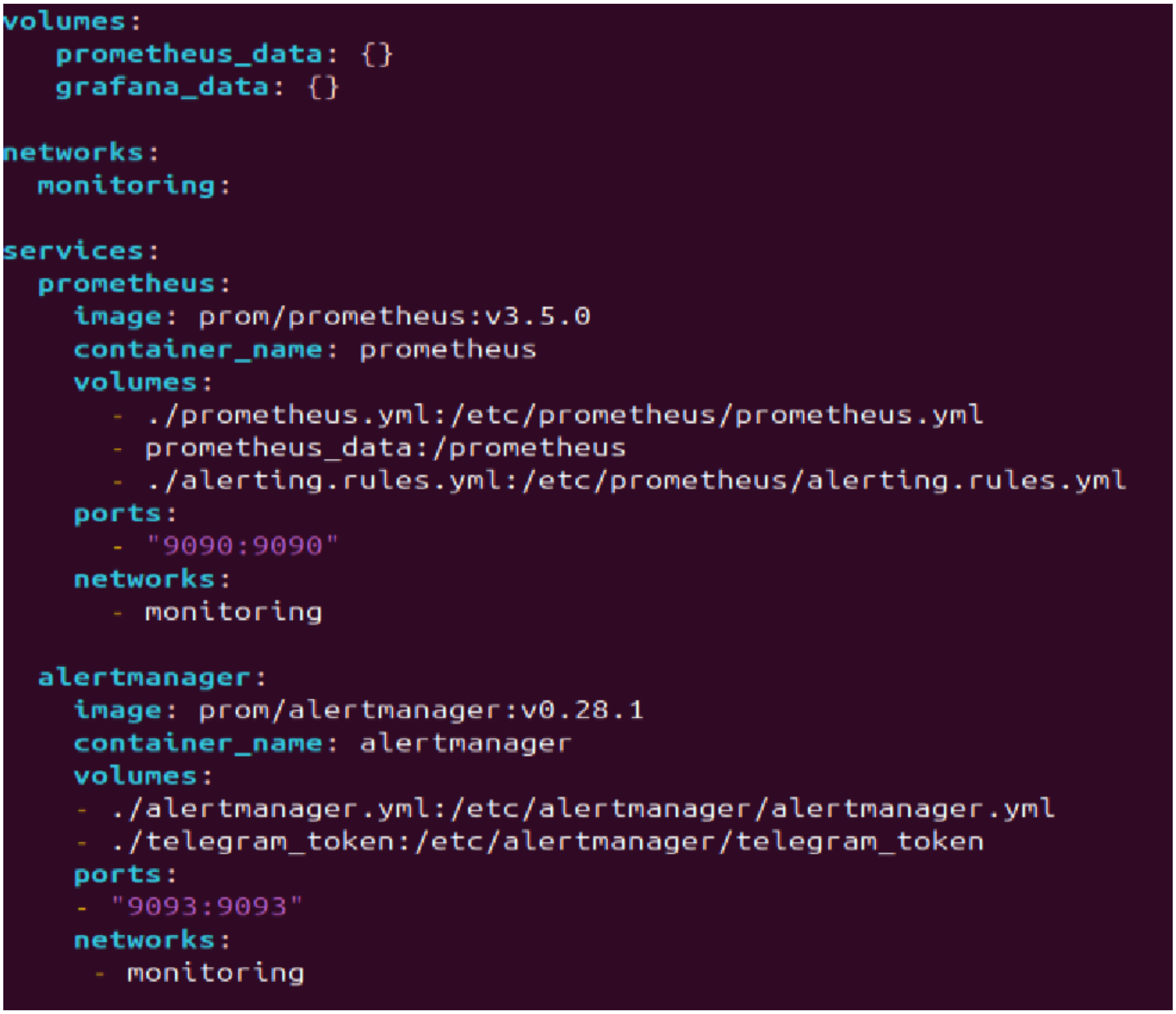
```
global:
  scrape_interval:     15s # By default, scrape targets every 15 seconds.

  # Attach these labels to any time series or alerts when communicating with
  # external systems (federation, remote storage, Alertmanager).
rule_files:
  - "alerting.rules.yml"

alerting:
  alert_relabel_configs:
  alertmanagers:
    - static_configs:
        - targets:
          - "alertmanager:9093"
# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
scrape_configs:
  # The job name is added as a label `job=<job_name>` to any timeseries scraped from this config.
  - job_name: 'prometheus'
    # Override the global default and scrape targets from this job every 5 seconds.
    scrape_interval: 5s
    static_configs:
      - targets: ['localhost:9090','node_exporter:9100']
~
~
~
~
~
~
```



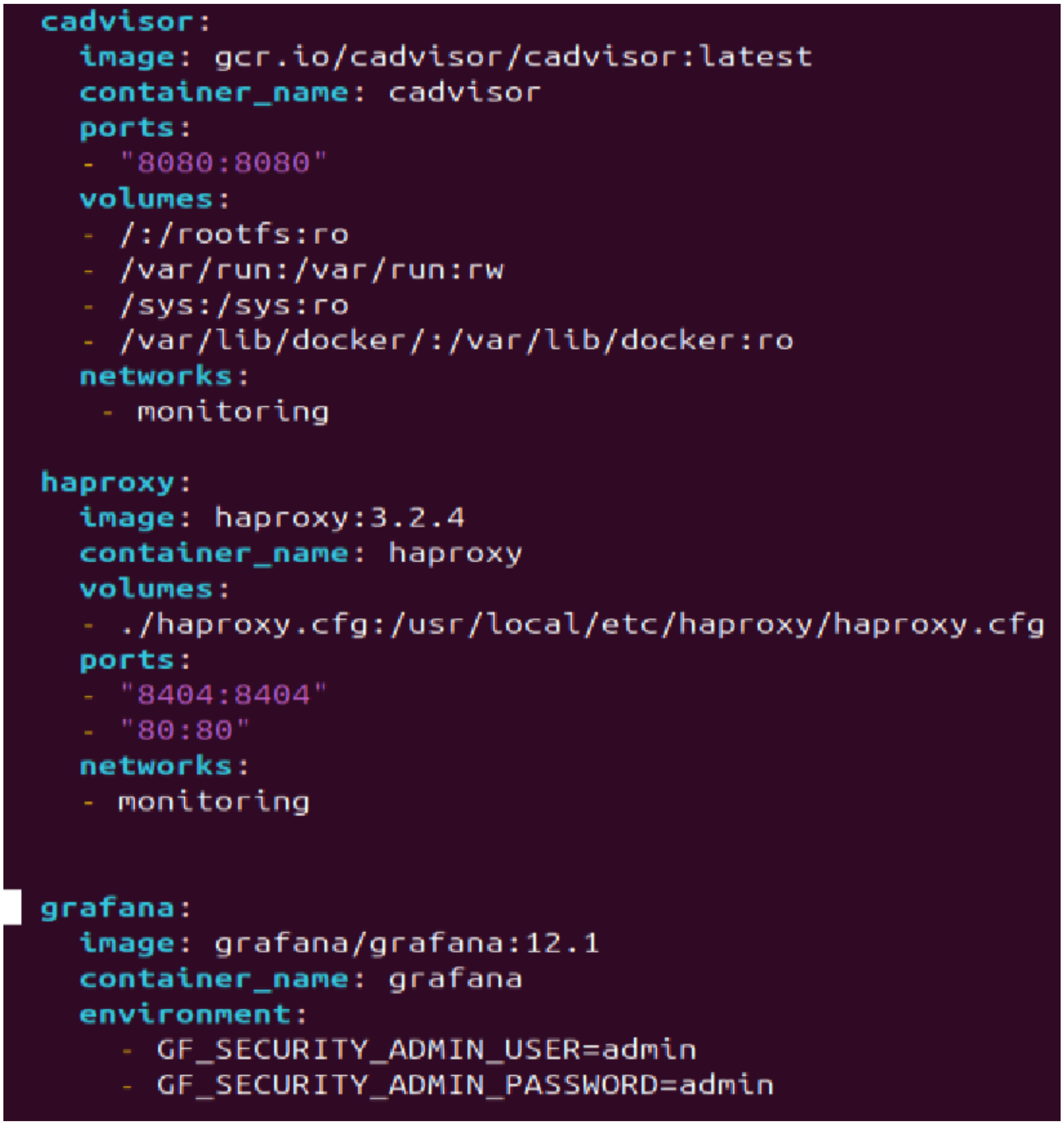
soha26_bot /newbot
Alert: Node Exporter is down on instancenode_exporter:9100
description:
Severity: page
Instance: node_exporter:9100
Starts At: 2025-09-16 23:57:28.761 +0000 UTC

11:57 PM

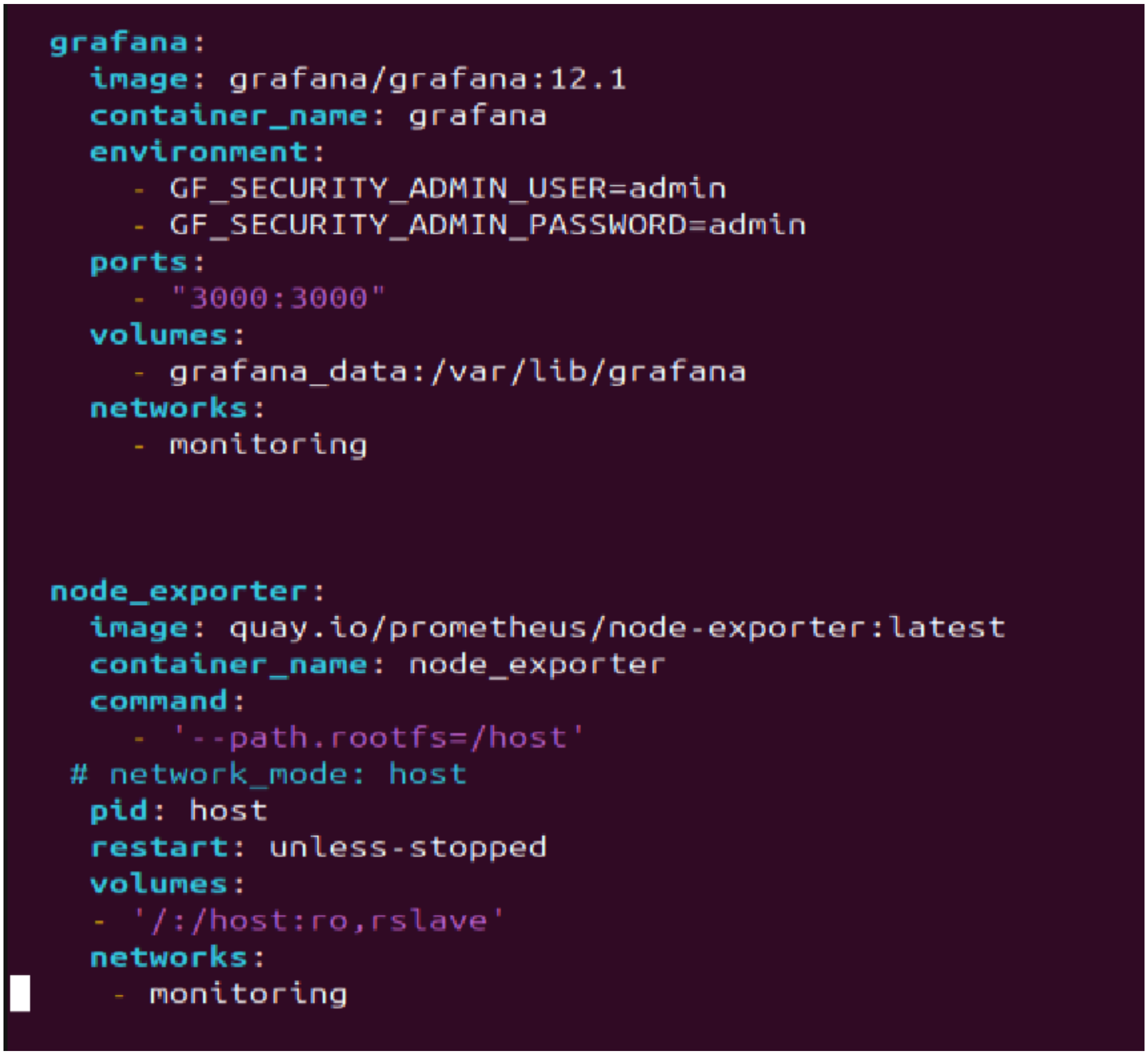
Task2:



```
volumes:
   prometheus_data: {}
   grafana_data: {}

networks:
  monitoring:

services:
  prometheus:
    image: prom/prometheus:v3.5.0
    container_name: prometheus
    volumes:
      - ./prometheus.yml:/etc/prometheus/prometheus.yml
      - prometheus_data:/prometheus
      - ./alerting.rules.yml:/etc/prometheus/alerting.rules.yml
    ports:
      - "9090:9090"
    networks:
      - monitoring

  alertmanager:
    image: prom/alertmanager:v0.28.1
    container_name: alertmanager
    volumes:
    - ./alertmanager.yml:/etc/alertmanager/alertmanager.yml
    - ./telegram_token:/etc/alertmanager/telegram_token
    ports:
    - "9093:9093"
    networks:
     - monitoring
```

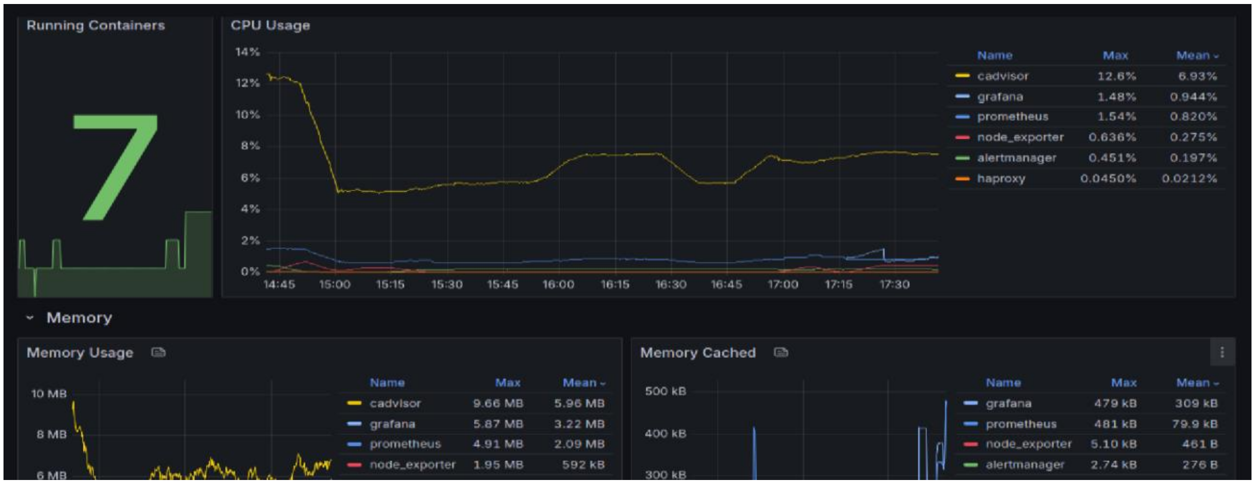


```
  cadvisor:
    image: gcr.io/cadvisor/cadvisor:latest
    container_name: cadvisor
    ports:
    - "8080:8080"
    volumes:
    - /:/rootfs:ro
    - /var/run:/var/run:rw
    - /sys:/sys:ro
    - /var/lib/docker/:/var/lib/docker:ro
    networks:
     - monitoring

  haproxy:
    image: haproxy:3.2.4
    container_name: haproxy
    volumes:
    - ./haproxy.cfg:/usr/local/etc/haproxy/haproxy.cfg
    ports:
    - "8404:8404"
    - "80:80"
    networks:
    - monitoring


  grafana:
    image: grafana/grafana:12.1
    container_name: grafana
    environment:
      - GF_SECURITY_ADMIN_USER=admin
      - GF_SECURITY_ADMIN_PASSWORD=admin
```



```
  grafana:
    image: grafana/grafana:12.1
    container_name: grafana
    environment:
      - GF_SECURITY_ADMIN_USER=admin
      - GF_SECURITY_ADMIN_PASSWORD=admin
    ports:
      - "3000:3000"
    volumes:
      - grafana_data:/var/lib/grafana
    networks:
      - monitoring



  node_exporter:
    image: quay.io/prometheus/node-exporter:latest
    container_name: node_exporter
    command:
      - '--path.rootfs=/host'
   # network_mode: host
    pid: host
    restart: unless-stopped
    volumes:
    - '/:/host:ro,rslave'
    networks:
     - monitoring
```

Cadvisor dashboard:

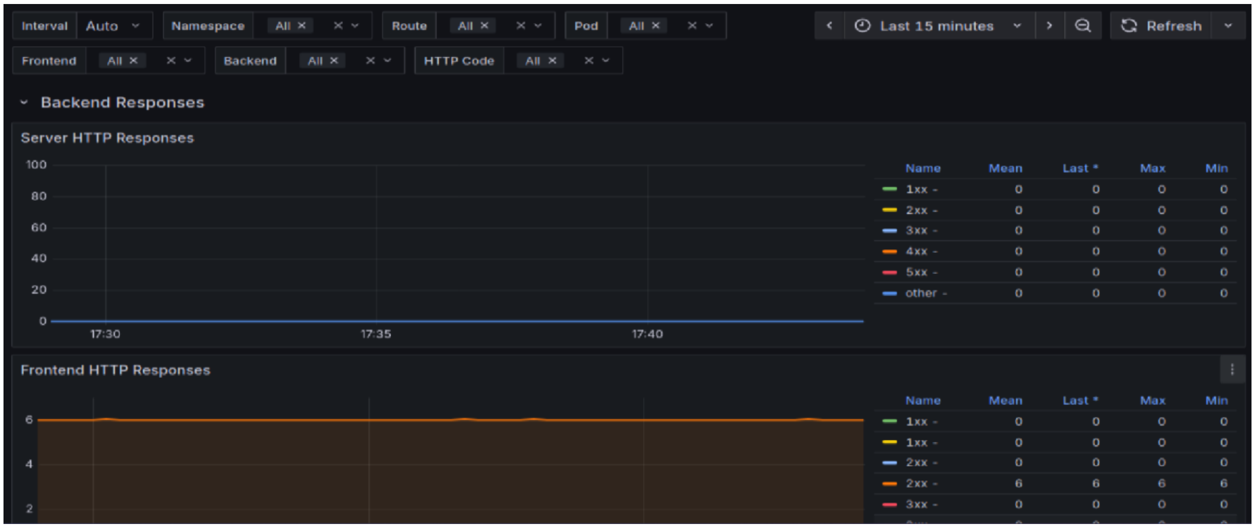


	Name	Max	Mean ⌄
🟨	cadvisor	12.6%	6.93%
🟦	grafana	1.48%	0.944%
🟦	prometheus	1.54%	0.820%
🟥	node_exporter	0.636%	0.275%
🟩	alertmanager	0.451%	0.197%
🟧	haproxy	0.0450%	0.0212%

	Name	Max	Mean ⌄
🟨	cadvisor	9.66 MB	5.96 MB
🟦	grafana	5.87 MB	3.22 MB
🟦	prometheus	4.91 MB	2.09 MB
🟥	node_exporter	1.95 MB	592 kB

	Name	Max	Mean ⌄
🟦	grafana	479 kB	309 kB
🟦	prometheus	481 kB	79.9 kB
🟥	node_exporter	5.10 kB	461 B
🟩	alertmanager	2.74 kB	276 B

Haproxy dashboard:



Interval: Auto | Namespace: All | Route: All | Pod: All | Last 15 minutes | Refresh

Frontend: All | Backend: All | HTTP Code: All

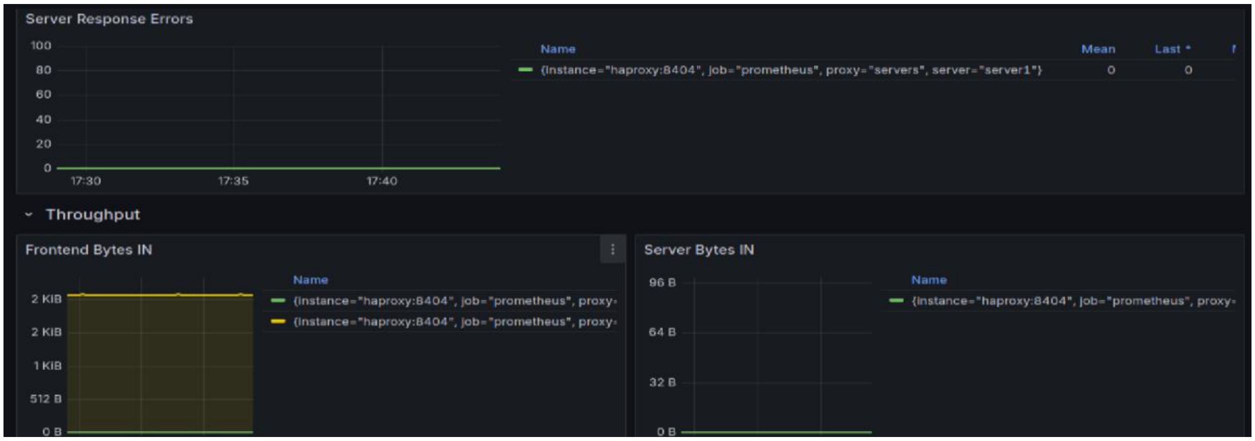
Backend Responses

Server HTTP Responses

	Name	Mean	Last *	Max	Min
🟩	1xx -	0	0	0	0
🟨	2xx -	0	0	0	0
🟦	3xx -	0	0	0	0
🟧	4xx -	0	0	0	0
🟥	5xx -	0	0	0	0
🟦	other -	0	0	0	0

Frontend HTTP Responses

	Name	Mean	Last *	Max	Min
🟩	1xx -	0	0	0	0
🟨	1xx -	0	0	0	0
🟦	2xx -	0	0	0	0
🟧	2xx -	6	6	6	6
🟥	3xx -	0	0	0	0



Server Response Errors

	Name	Mean	Last *
🟩	{instance="haproxy:8404", job="prometheus", proxy="servers", server="server1"}	0	0

Throughput

Frontend Bytes IN

	Name
🟩	{instance="haproxy:8404", job="prometheus", proxy=
🟨	{instance="haproxy:8404", job="prometheus", proxy=

Server Bytes IN

	Name
🟩	{instance="haproxy:8404", job="prometheus", proxy=