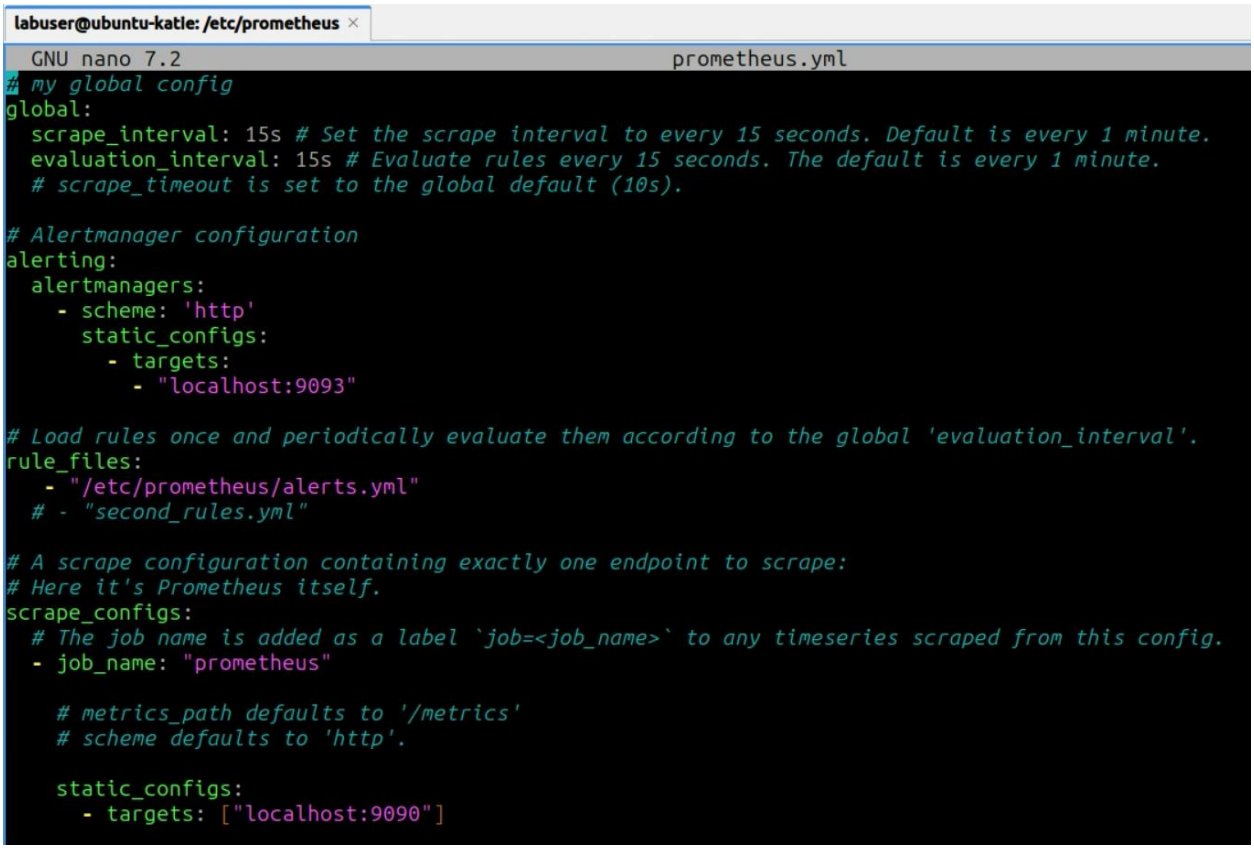
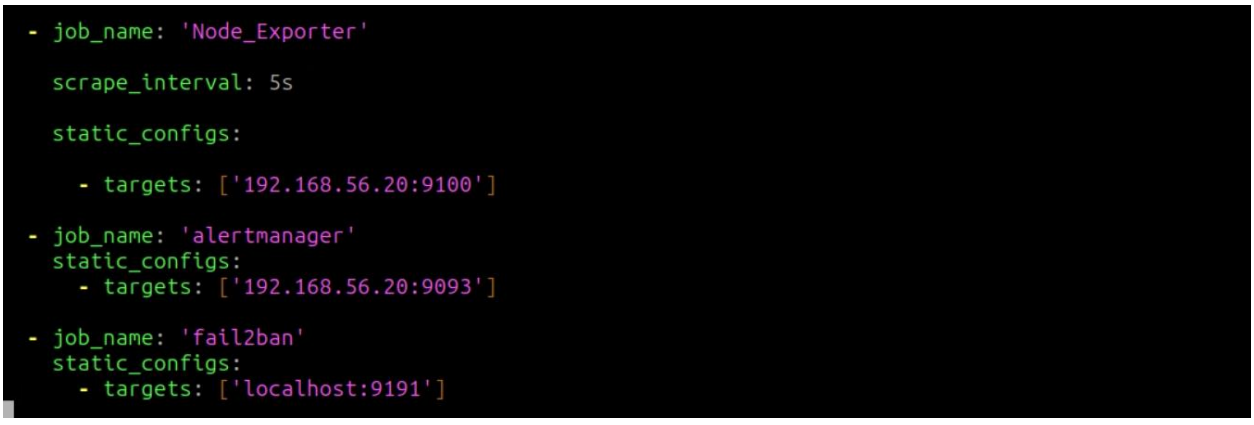
Systems Infrastructure and Security Project

To start this project, the first thing I did was install and configure Prometheus and Grafana. A Prometheus user had to be created and given the relevant permissions and then installed.

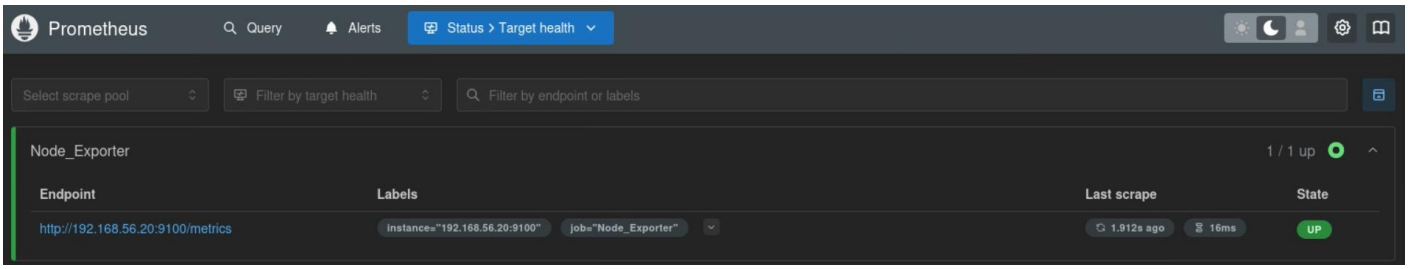




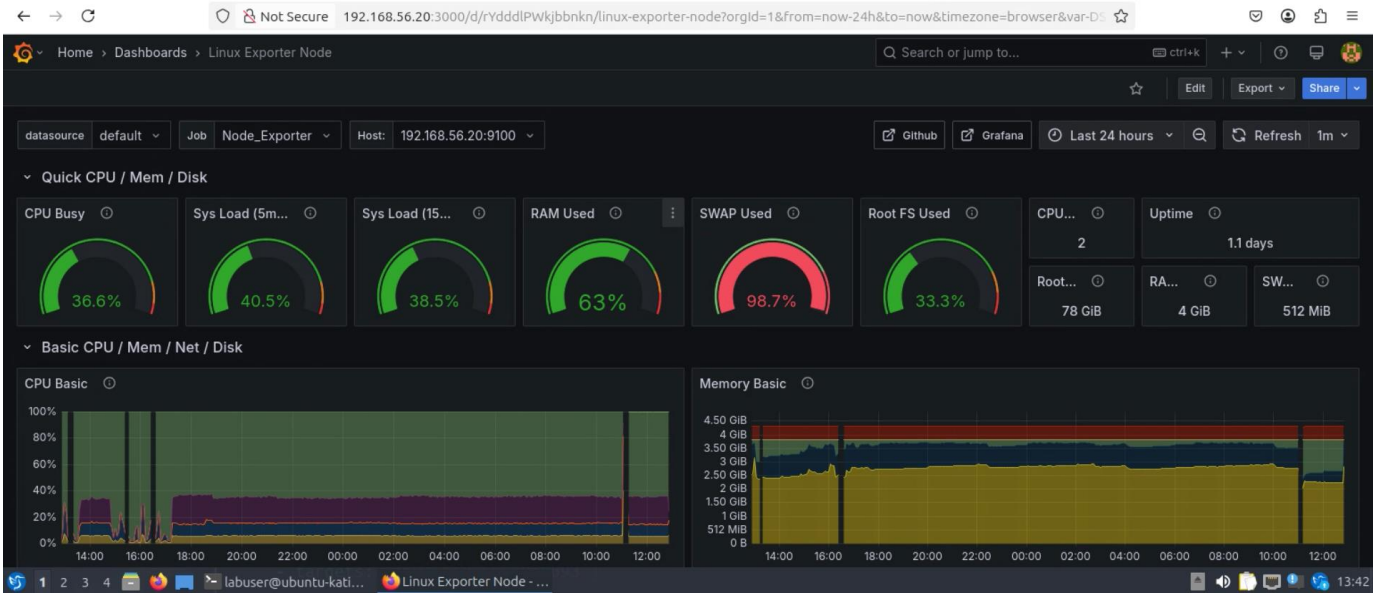
The prometheus.yml configuration file, edited to include exporters such as Node Exporter and fail2ban. Prometheus is now accessible on the web browser through http://192.168.56.20:9090

I then installed the latest version of Node Exporter, to use as a Prometheus target. After this was successfully installed, I added it into the prometheus.yml file, as seen above.

Prometheus can now scrape metrics from Node Exporter.

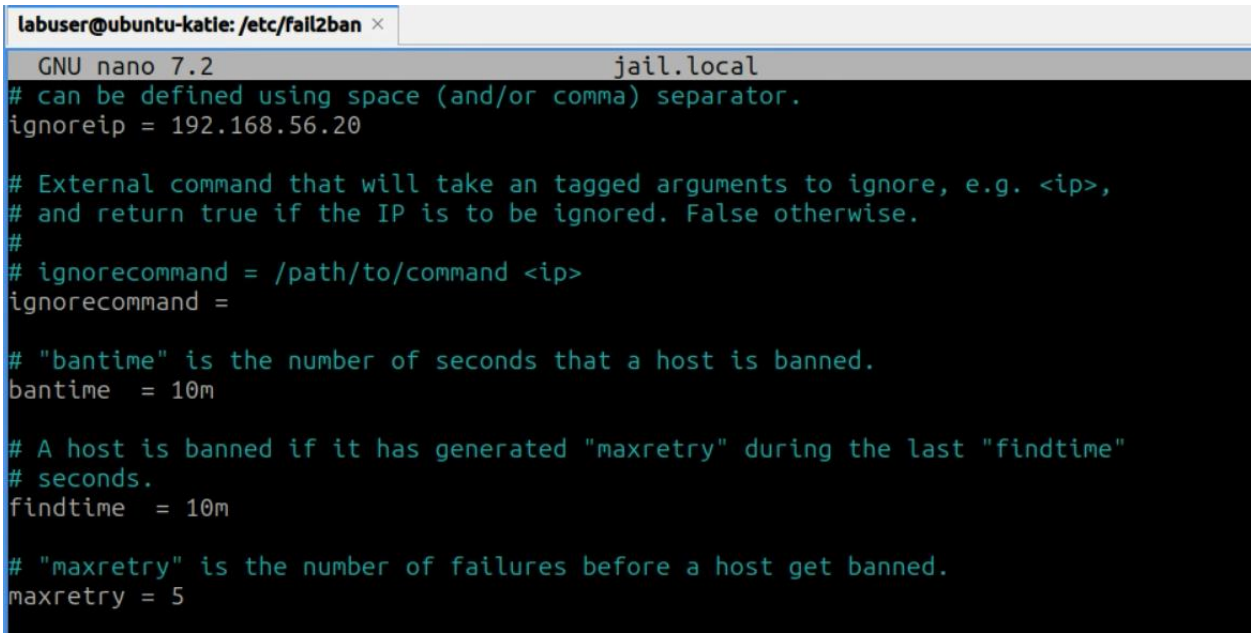


Next step was installing grafana. After this was completed, I added Prometheus as a data source for an dashboards I might add. I used a premade template for my first dashboard, which uses Node Exporter to monitor system metrics such as CPU and memory usage.

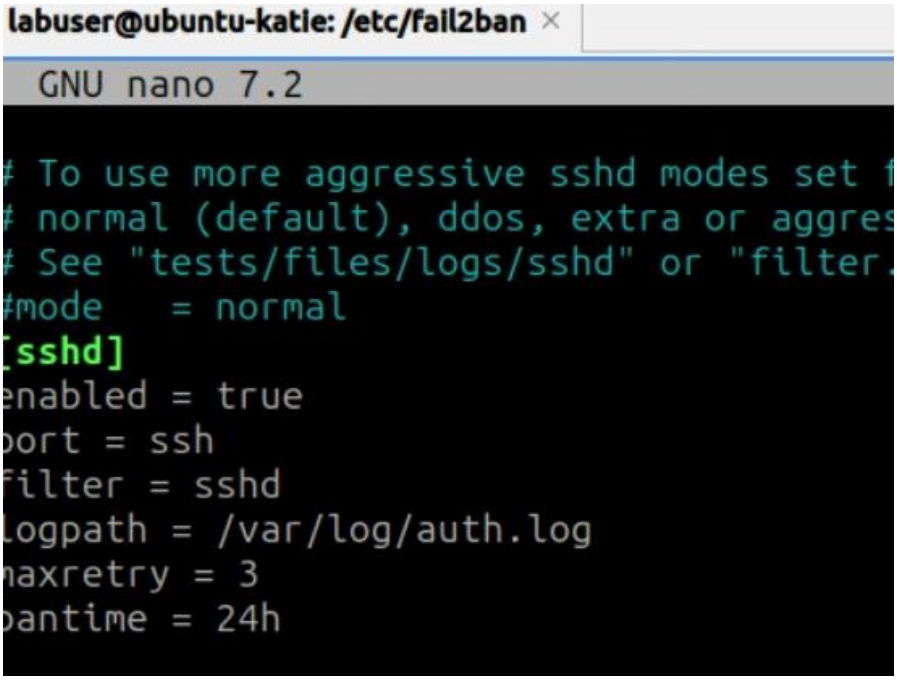


Fail2Ban is an intrusion protection service to protect your servers from brute force attacks and unwanted threats. I installed this on my system to monitor repeated SSH login attempts and block any suspicious IPs.

After installing Fail2Ban using apt, I copied the jail.conf file to a new file called jail.local to create local configurations without changing the original file.

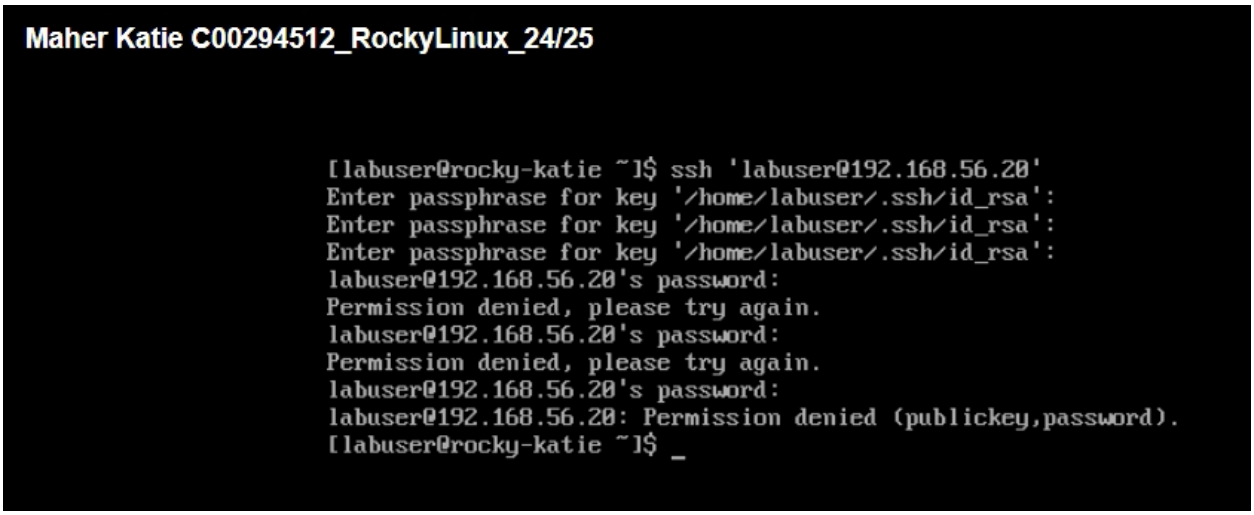


I configured fail2ban to ignore my IP address, so I don't ban myself by mistake, and then to ban a host for 10 minutes if they retry (etc wrong login attempts) over 5 times.

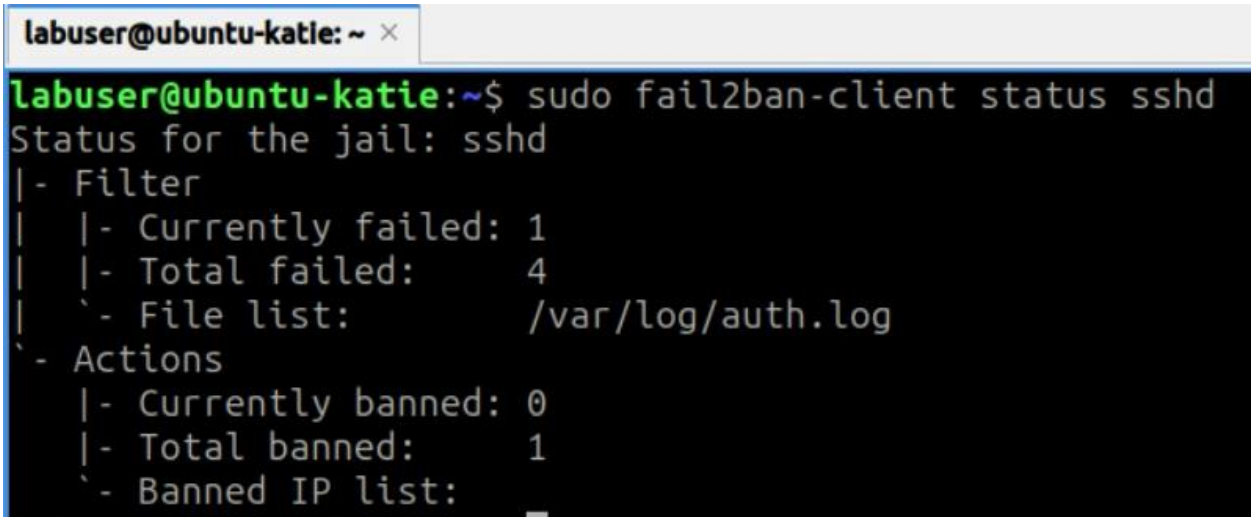


If a user tries to log in via ssh more than three times incorrectly, their IP address is banned for 24 hours to prevent brute force attacks.

To test this, I set up an SSH connection between the Rocky Linux server and Ubuntu machine. I then did three incorrect login attempts to ensure that the address would be banned.

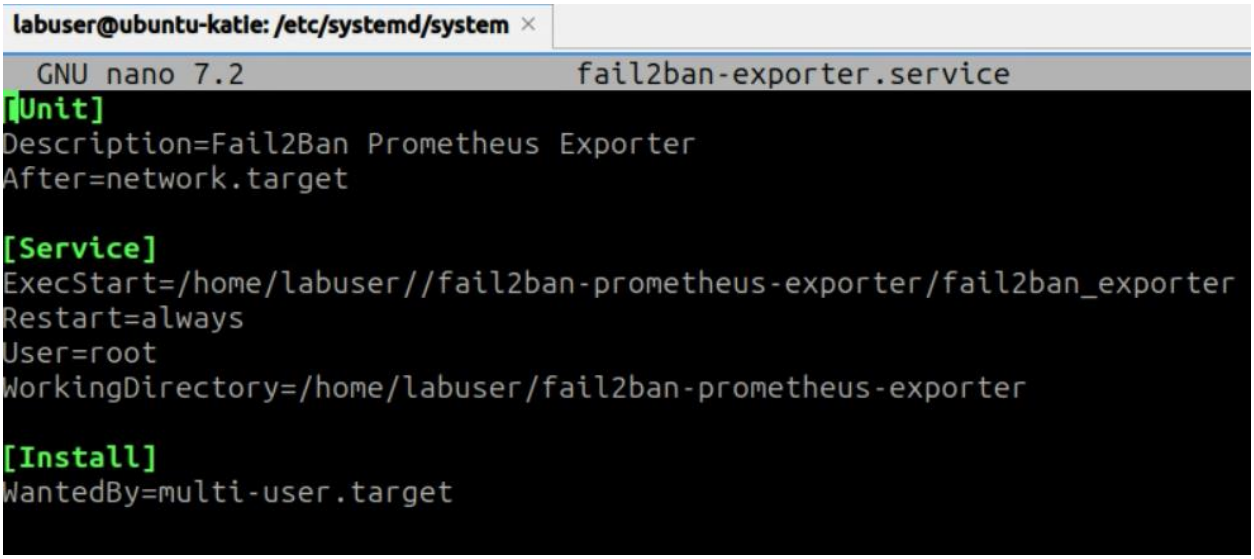


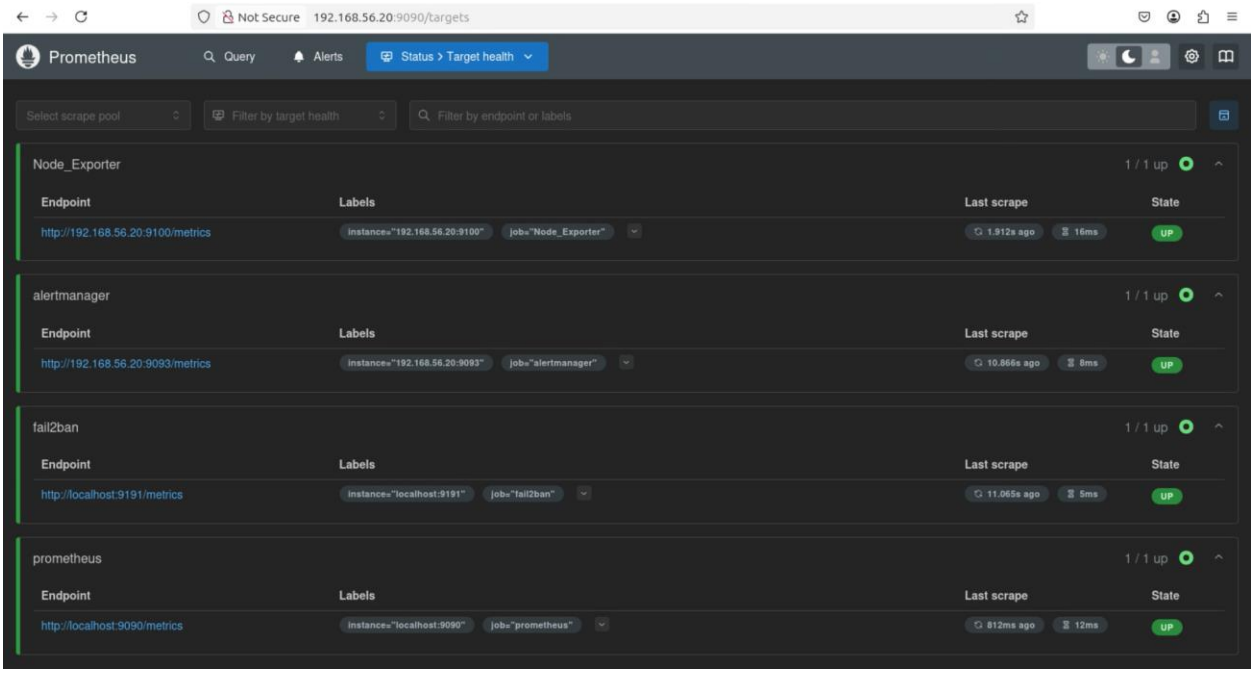
After the failed login attempts, fail2ban adds this IP to it's 'jail' for 24 hours.



To use fail2ban as a data source for Prometheus, I cloned and installed a Fail2Ban exporter from GitHub. I had to install GO, as this was the language the exporter was written in and then add the exporter to the systemd directory with its own service file, so that the binary

would run on startup and Prometheus could constantly scrape metrics.





Prometheus can now see all its data sources, which are all up.

To notify administrators of any important events, such as failed login attempts or a service being down, I used Prometheus Alertmanager. After doing the initial install, I had to edit the prometheus.yml file to include different alerting rules and what file these rules would be found in.

```
# Alertmanager configuration
alerting:
  alertmanagers:
    - scheme: 'http'
      static_configs:
        - targets:
          - "localhost:9093"

# Load rules once and periodically evaluate them according to the global 'evaluation_interval'.
rule_files:
   - "/etc/prometheus/alerts.yml"
  # - "second_rules.yml"
```

This part of the configuration file points to a file called alerts.yml, which defines the rules

```
 GNU nano 7.2                                alerts.yml
groups:
  - name: node_exporter_alerts
    rules:
      - alert: NodeExporterDown
        expr: up{job="Node_Exporter"} == 0
        for: 1m
        labels:
          severity: critical
        annotations:
          summary: "Node Exporter down"
          description: "Node Exporter has been down for more than 1 minute."

  - name: fail2ban_alerts
    rules:
      - alert: Fail2BanBannedIP
        expr: f2b_jail_banned_current > 0
        for: 1m
        labels:
          severity: critical
        annotations:
          summary: "Fail2Ban Alert: IP Banned"
          description: "Fail2Ban has banned an IP. Current bans: {{ $value }}"
```

I have rules for two alerts, one if Node Exporter has been down for more than one minute, and one for if Fail2Ban has banned an IP. These alerts work by using Prometheus's query language PromQL to check for metrics such as if a job is up, or if the number of bans is greater than 0.

To actually send the alerts, I initially chose email alerts, but there were issues with authentication, so I decided to use Discord to send alerts. The alert method is specified in the alertmanager.yml file.

```
global:
  resolve_timeout: 5m  # Optional timeout setting

route:
  group_by: ['alertname','job']
  group_wait: 30s
  group_interval: 5m
  repeat_interval: 1h
  receiver: 'discord-alerts'

receivers:
  - name: 'discord-alerts'
    webhook_configs:
      - url: 'http://localhost:9095'
        send_resolved: true
```

I used a docker configuration called alertmanager-discord that I found on GitHub to process these alerts effectively. This runs on port 9095 (originally 9094 but I had alertmanager using that port).

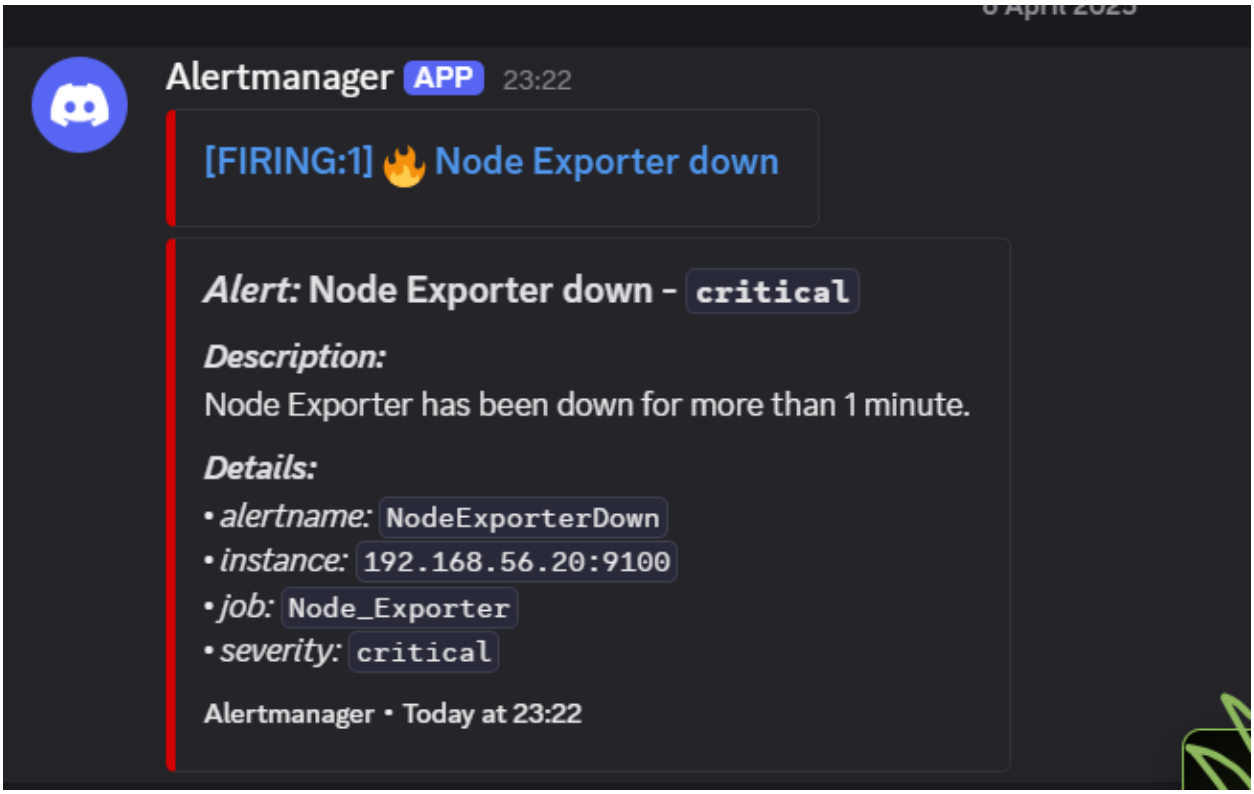
This uses a custom webhook to send alerts to a server and channel of your choice.

The next step was to put it all together and test if the alerts were sent correctly.

To test the alerts, I stopped the Node Exporter service, so this should notify alertmanager.

Prometheus is now ‘firing’, meaning the query has been met and Node Exporter has been down for more than a minute.

Alertmanager has received and attempted to send the alert to the discord channel I have specified.



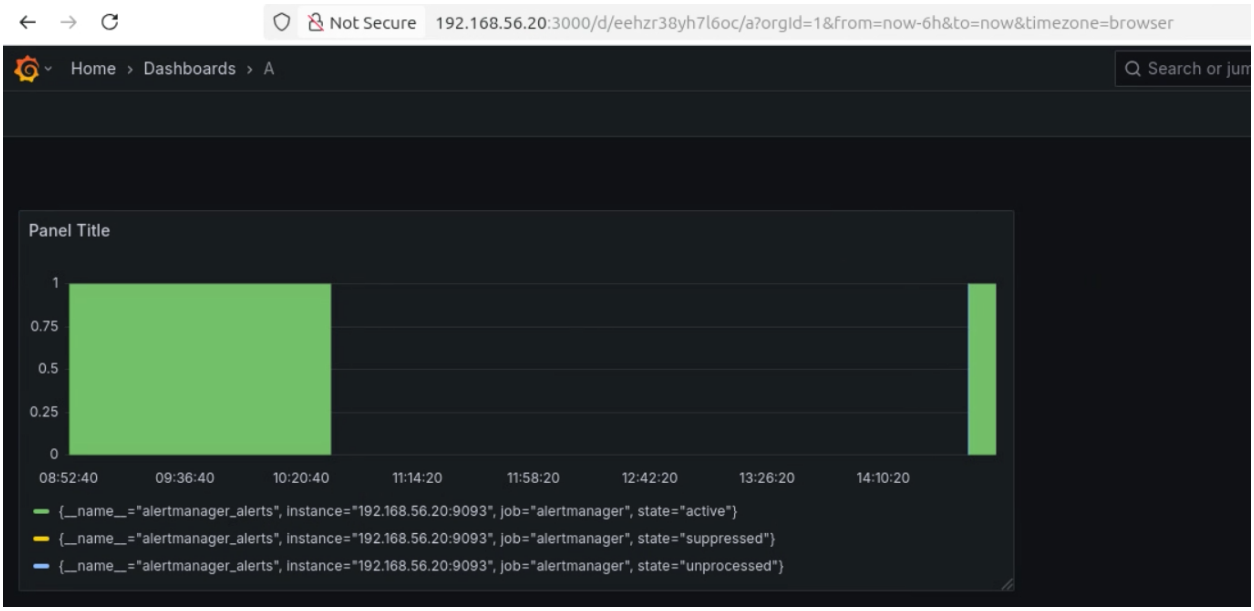
Discord successfully recieves the alert and I can see it!

I then went back to Grafana to configure my other two dashboards. I chose to do one using Fail2Ban that shows how many current bans there are.

In this screenshot it goes from one to zero, as the 24 hour ban time had expired.

My other dashboard uses Alertmanager to show how many current alerts there are.

As you can see in the below screenshot, it was at 1 for a time because of the banned IP, but at zero when that ban expired. It then went back up to one when I stopped Node Exporter for demonstration purposes.



Resources used:

https://runcloud.io/blog/what-is-fail2ban

https://grafana.com/blog/2020/02/25/step-by-step-guide-to-setting-up-prometheus-alertmanager-with-slack-pagerduty-and-gmail/

https://github.com/hctrdev/fail2ban-prometheus-exporter

https://www.fosstechnix.com/install-prometheus-and-grafana-on-ubuntu-24-04/

https://github.com/prometheus/prometheus/releases/download/v2.46.0/prometheus-2.46.0.linux-amd64.tar.gz

https://github.com/rogerrum/alertmanager-discord