

# **Systems Infrastructure and Security Project**

---

Adam Romanowicz

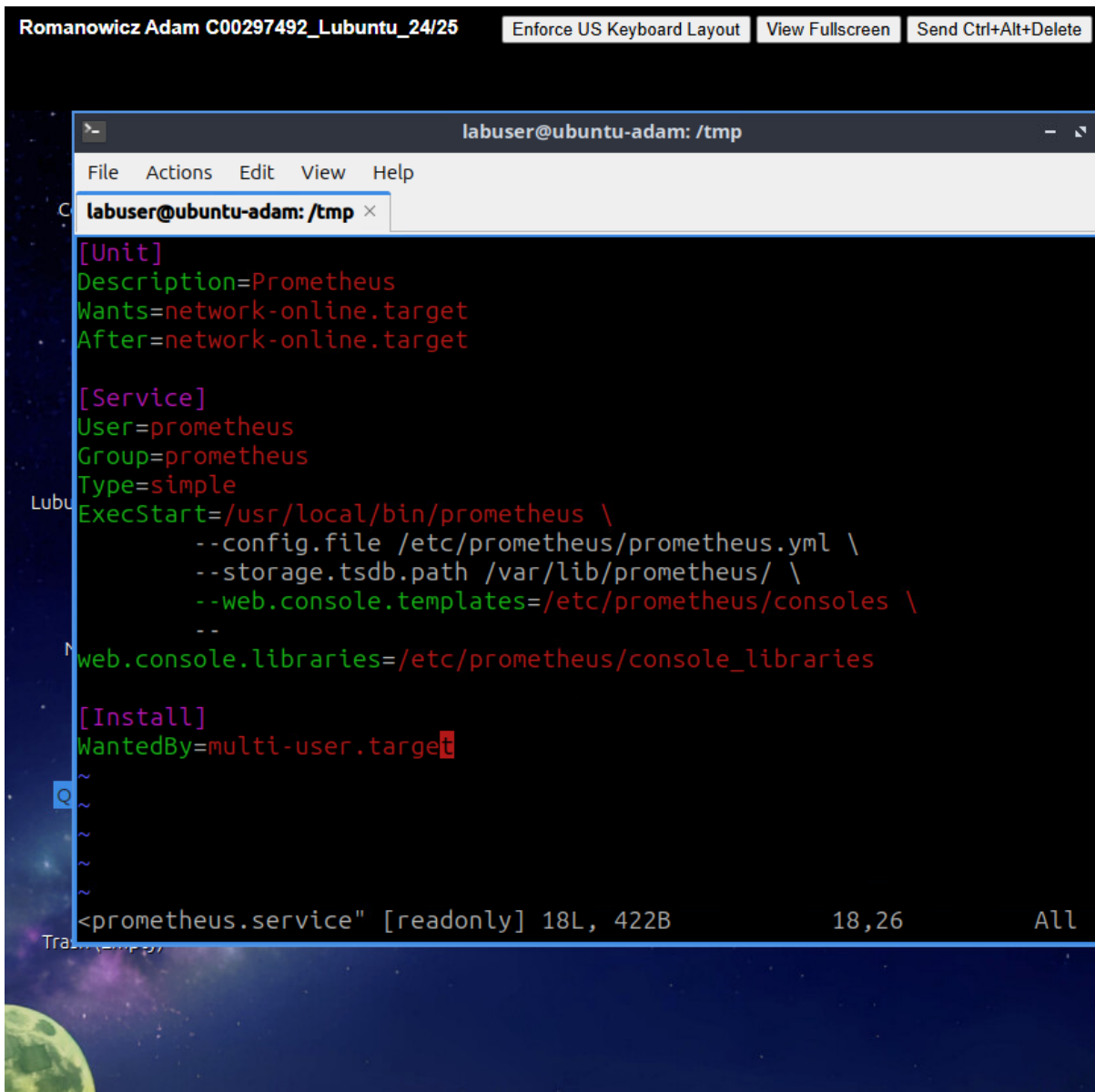
C00297492

IT Management

## **Prometheus**

Prometheus was installed to gather information about the system and alert administrators of potential security events as they occur. To setup Prometheus you first make a user called Prometheus with no home directory and with a shell of /bin/false. Make a directory in /etc called Prometheus and a directory in /var/lib called the same. Set the ownership of /var/lib/Prometheus to the newly created Prometheus user. Download the binary file for Prometheus from the github using wget and get the newest release. Unzip Prometheus using tar then move the console files into /etc/Prometheus and move the Prometheus.yml config file to the same location. Set the ownership of these files and folder to the Prometheus user. Move the Prometheus file from the zip folder into usr/local/bin/ and set Prometheus as its owner. Edit the configuration Prometheus.yml file and make sure the static target is either the ip of your desired host or in this case you can use localhost as the Prometheus GUI is running on the same system as Prometheus itself. Here is the system service file that runs Prometheus so it does not have to be manually ran on startup and can easily be disabled. A separate user was created so it can run the service independently. Enter the GUI by putting

<http://localhost:9090> in a web browser.



```
labuser@ubuntu-adam: /tmp
[Unit]
Description=Prometheus
Wants=network-online.target
After=network-online.target

[Service]
User=prometheus
Group=prometheus
Type=simple
ExecStart=/usr/local/bin/prometheus \
    --config.file /etc/prometheus/prometheus.yml \
    --storage.tsdb.path /var/lib/prometheus/ \
    --web.console.templates=/etc/prometheus/consoles \
    --web.console.libraries=/etc/prometheus/console_libraries

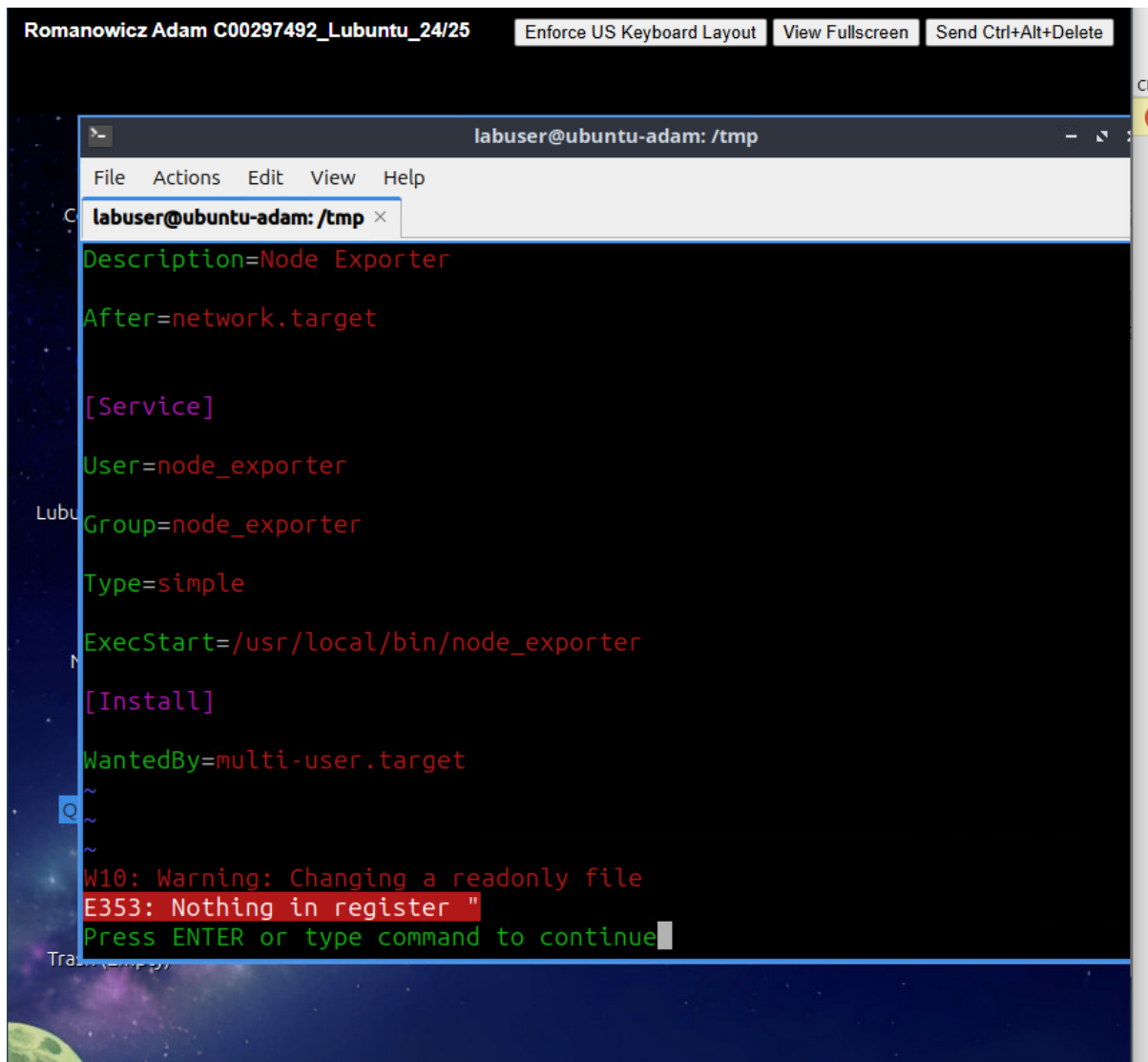
[Install]
WantedBy=multi-user.target

<prometheus.service" [readonly] 18L, 422B      18,26      All
```

/etc/systemd/system/Prometheus.service

Three exporters were installed for Prometheus to expose certain metrics; Node exporter for basic system information such as CPU, memory and disk usage which provides system health data to administrators e.g. high cpu usage could be a sign of malware operating on the system which would hinder server operations; Blackbox exporter to show http traffic and health which would inform administrators of a DoS attack if a high amount of traffic is recorded; and Fail2ban exporter to show how many ips have been banned, failed login attempts, etc which informs administrators of a brute force attack if a high amount of failed login attempts and multiple ips banned within a short time frame. Below the configurations and service files are included. To setup node exporter wget it off the github and download the latest release. Unzip it using tar and move the binary file to /usr/local/bin/. Create a user called node\_exporter and make it have no

home directory with the shell being /bin/false. Create a service file in /etc/systemd/system. Add node exporter into the Prometheus.yml configuration file with its static target as localhost:9100. To setup blackbox exporter wget it from the github and get the latest version. Unzip the folder using tar and move blackbox\_exporter to /usr/local/bin. Create a blackbox directory in .etc and move blackbox.yml config file to that folder. Create a user called blackbox with no home directory and the shell /bin/false. Set ownership permissions to that blackbox user on the /usr/local/bin/blackbox\_exporter file and on the /etc/blackbox directory and its contents. Create the service file in /etc/systemd/system. Edit Prometheus.yml and add localhost:9115 to static\_configs under the Prometheus job then make a new job. This config is for http probing and make sure the target is the server and the replacement is the blackbox ip of 127.0.0.1:9115. To setup fail2ban exporter download it off the github and move the fail2ban\_exporter to /usr/sbin. Create a service file in /etc/systemd/system. Then create a fail2ban exporter job in Prometheus.yml.



```
Romanowicz Adam C00297492_Lubuntu_24/25  Enforce US Keyboard Layout  View Fullscreen  Send Ctrl+Alt+Delete
labuser@ubuntu-adam: /tmp
File  Actions  Edit  View  Help
labuser@ubuntu-adam: /tmp x
Description=Node Exporter
After=network.target

[Service]
User=node_exporter
Group=node_exporter
Type=simple
ExecStart=/usr/local/bin/node_exporter

[Install]
WantedBy=multi-user.target

W10: Warning: Changing a readonly file
E353: Nothing in register "
Press ENTER or type command to continue
```

/etc/systemd/system/node\_exporter.service

Romanowicz Adam C00297492\_Lubuntu\_24/25   Enforce US Keyboard Layout   View Fullscreen   Send Ctrl+Alt+Delete

labuser@ubuntu-adam: /tmp

File   Actions   Edit   View   Help

labuser@ubuntu-adam: /tmp ×

```
rule_files:
  # - "first_rules.yml"
  # - "second_rules.yml"

# 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 timeserie
  # scraped from this config.
  - job_name: "prometheus"

    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.

    static_configs:
      - targets: ["localhost:9090"]

      - job_name: "Node_Exporter"

    scrape_interval: 5s

    static_configs:
      - targets: ["192.168.56.20:9100"]
<us/prometheus.yml" [readonly] 36L, 1051B   35,19   Bot
```

/etc/prometheus/prometheus.yml

```
Romanowicz Adam C00297492_Lubuntu_24/25

labuser@ubuntu-adam: /lib/systemd/system
File Actions Edit View Help
labuser@ubuntu-adam: /lib/systemd/system x
[Unit]
Description=Blackbox Exporter Service
Wants=network-online.target
After=network-online.target

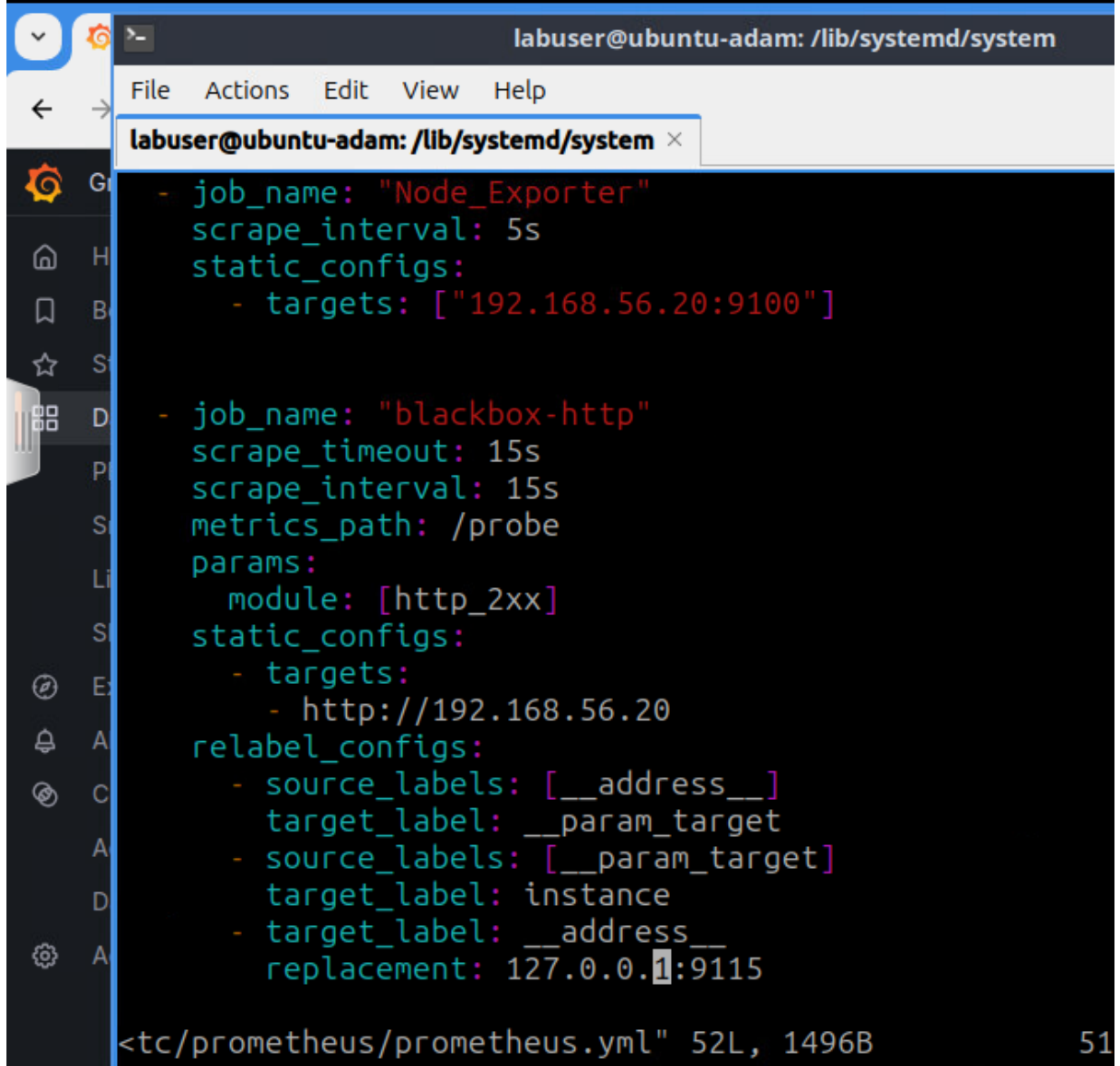
[Service]
Type=simple
User=blackbox
Group=blackbox
ExecStart=/usr/local/bin/blackbox_exporter \
--config.file=/etc/blackbox/blackbox.yml \
--web.listen-address=":9115"
Restart=always

[Install]
WantedBy=multi-user.target

"blackbox.service" 16L, 325B 16,26 All
```

/etc/systemd/system/blackbox.service

Scrape interval refers to how long should the exporter wait till it updates its values from the service it is taking metrics from. Scrape timeout refers to how long it should wait till it stops trying to scrape a metric if it cannot find the value (e.g. the service could be disabled).



The screenshot shows a terminal window with a dark background and light-colored text. The window title is 'labuser@ubuntu-adam: /lib/systemd/system'. The terminal displays the configuration for two Prometheus jobs. The first job is 'Node\_Exporter' with a scrape interval of 5s and a static config target of '192.168.56.20:9100'. The second job is 'blackbox-http' with a scrape timeout of 15s, a scrape interval of 15s, and a metrics path of '/probe'. It uses the 'http\_2xx' module and has three relabel configs: one for the target label to '\_\_param\_target', one for the target label to 'instance', and one for the target label to '\_\_address\_\_' with a replacement of '127.0.0.1:9115'. The bottom of the terminal shows the file path '<tc/prometheus/prometheus.yml' with a size of 52L, 1496B and a line number of 51.

```
labuser@ubuntu-adam: /lib/systemd/system
- job_name: "Node_Exporter"
  scrape_interval: 5s
  static_configs:
    - targets: ["192.168.56.20:9100"]

- job_name: "blackbox-http"
  scrape_timeout: 15s
  scrape_interval: 15s
  metrics_path: /probe
  params:
    module: [http_2xx]
  static_configs:
    - targets:
      - http://192.168.56.20
  relabel_configs:
    - source_labels: [__address__]
      target_label: __param_target
    - source_labels: [__param_target]
      target_label: instance
    - target_label: __address__
      replacement: 127.0.0.1:9115

<tc/prometheus/prometheus.yml" 52L, 1496B 51
```

/etc/prometheus/prometheus.yml

Romanowicz Adam C00297492\_Lubuntu\_24/25 Enforce US Keyboard Layout View Fullscreen

F2B - Dashboards - Grafana Prometheus Time Series

labuser@ubuntu-adam: /lib/systemd/system

```
[Unit]
Description=Fail2ban metric exporter for Prometheus
Requires=network-online.target
After=network-online.target

[Service]
ExecStart=/usr/sbin/fail2ban_exporter
Restart=on-failure
RestartSec=5s
NoNewPrivileges=true

User=root
Group=root

[Install]
WantedBy=multi-user.target
~
~
~
~
~
~
"fail2ban_exporter.service" 16L, 281B 16,26 All
```

Name	Used	Name
sshd (localhost:9191)	4	sshd (localhost:9191)

/etc/systemd/system/fail2ban\_exporter.service

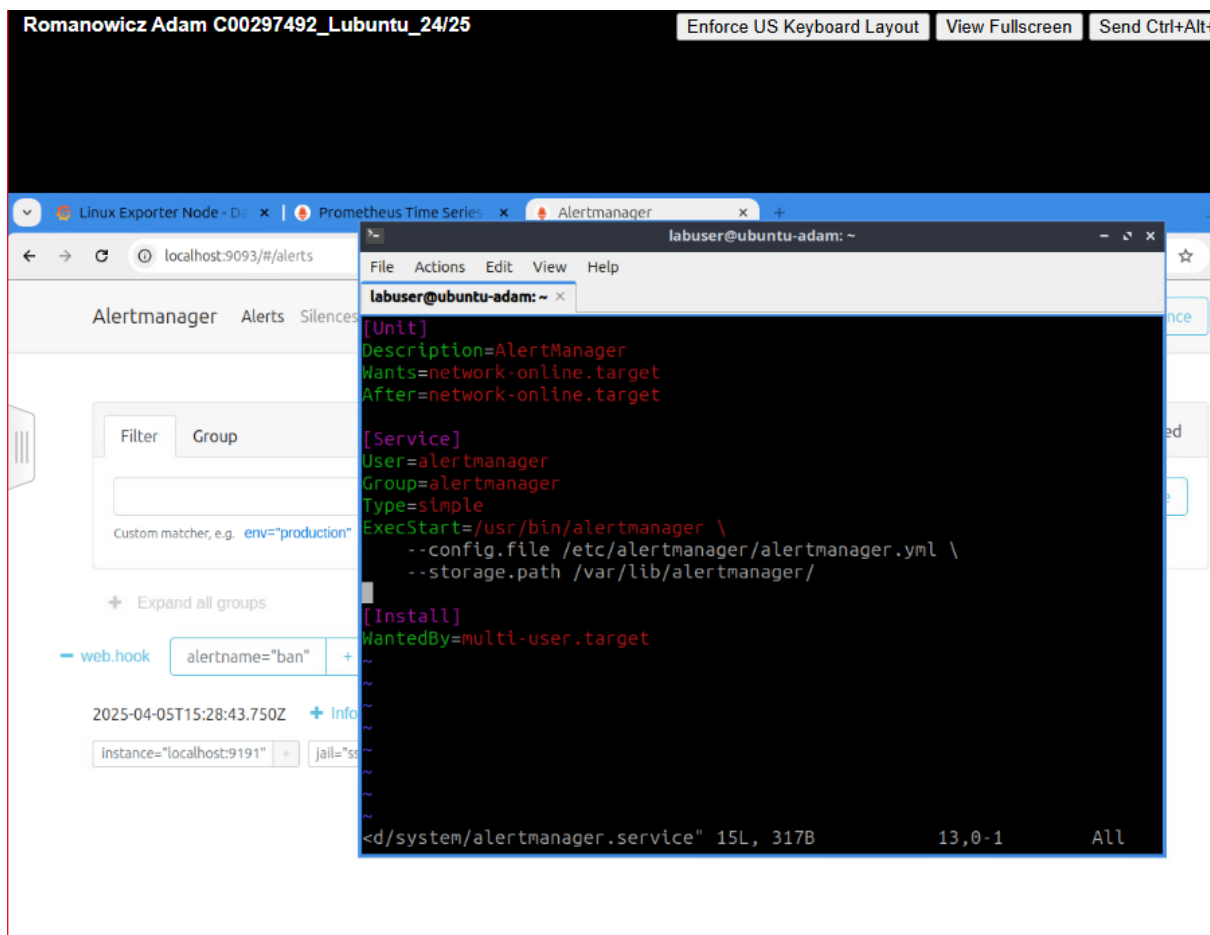


```
Romanowicz Adam C00297492_Lubuntu_24/25 Enforce US Keyboard Layout
F2B - Dashboards - Grafana Prometheus Time Series
labuser@ubuntu-adam: ~
File Actions Edit View Help
labuser@ubuntu-adam: ~ x
- job_name: "blackbox-http"
  scrape_timeout: 15s
  scrape_interval: 15s
  metrics_path: /probe
  params:
    module: [http_2xx]
  static_configs:
    - targets:
      - http://192.168.56.20
  relabel_configs:
    - source_labels: [__address__]
      target_label: __param_target
    - source_labels: [__param_target]
      target_label: instance
    - target_label: __address__
      replacement: 127.0.0.1:9115
- job_name: "fail2ban-exporter"
  scrape_timeout: 15s
  scrape_interval: 15s
  static_configs:
    - targets: ["localhost:9191"]
<tc/prometheus/prometheus.yml" 58L, 1636B 54,23
```

/etc/Prometheus/Prometheus.yml

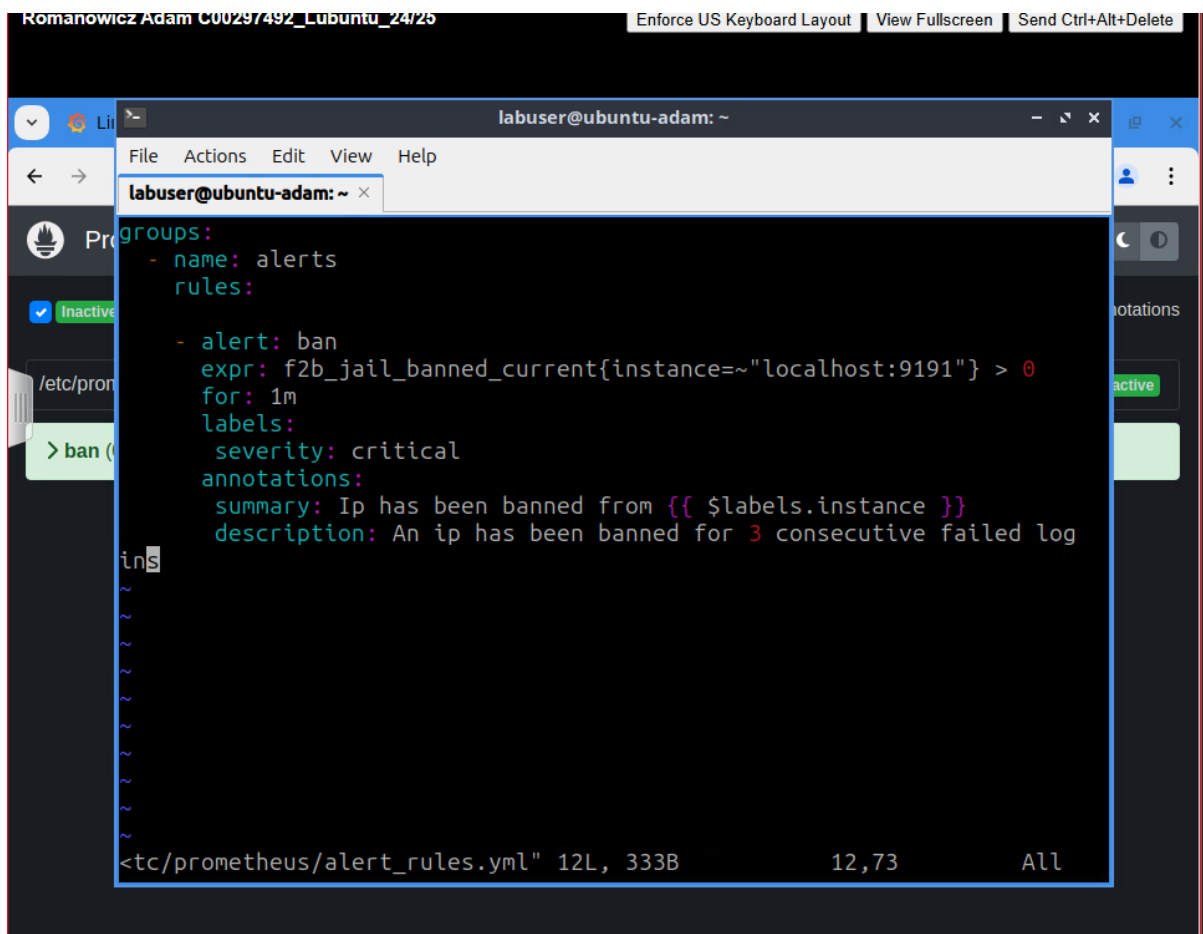
Alertmanager was also set up which groups alerts specified on a certain metric criteria and sends that alert to a administrator (in this case it is set up to send the alert via Slack channel). 2 alerts were configured which include High CPU usage which alerts administrators of average CPU usage above 80% for 1 minute indicating potential malicious code running on the server to disrupt server use; and a ban alert alerting administrators that an ip has been banned potentially indicating that an unauthorized agent tried to gain access to the server. To setup alertmanager download it from the github using wget. Create a alertmanager user with no home directory and the shell being /bin/false. Create a alertmanager directory in /var/lib and set ownership to alertmanager user in /etc/alertmanager and /var/lib/alertmanager. Unzip the tar file and

move alertmanager file to /usr/bin. Move the configuration file alertmanager.yml to /etc/alertmanager and set the owner to alertmanager. Create the service file in /usr/lib/systemd/system. Create alerts by creating an alert.yml file in /etc/Prometheus then add that file to the Prometheus.yml config file. Create a slack account then edit a channel and add incoming webhooks to it. Copy the url and add it to the alertmanager.yml receiver section.



/etc/systemd/system/alertmanager.service

This alert checks if the current amount of banned ips are greater than 0 and it waits 1 minute before firing the alert.



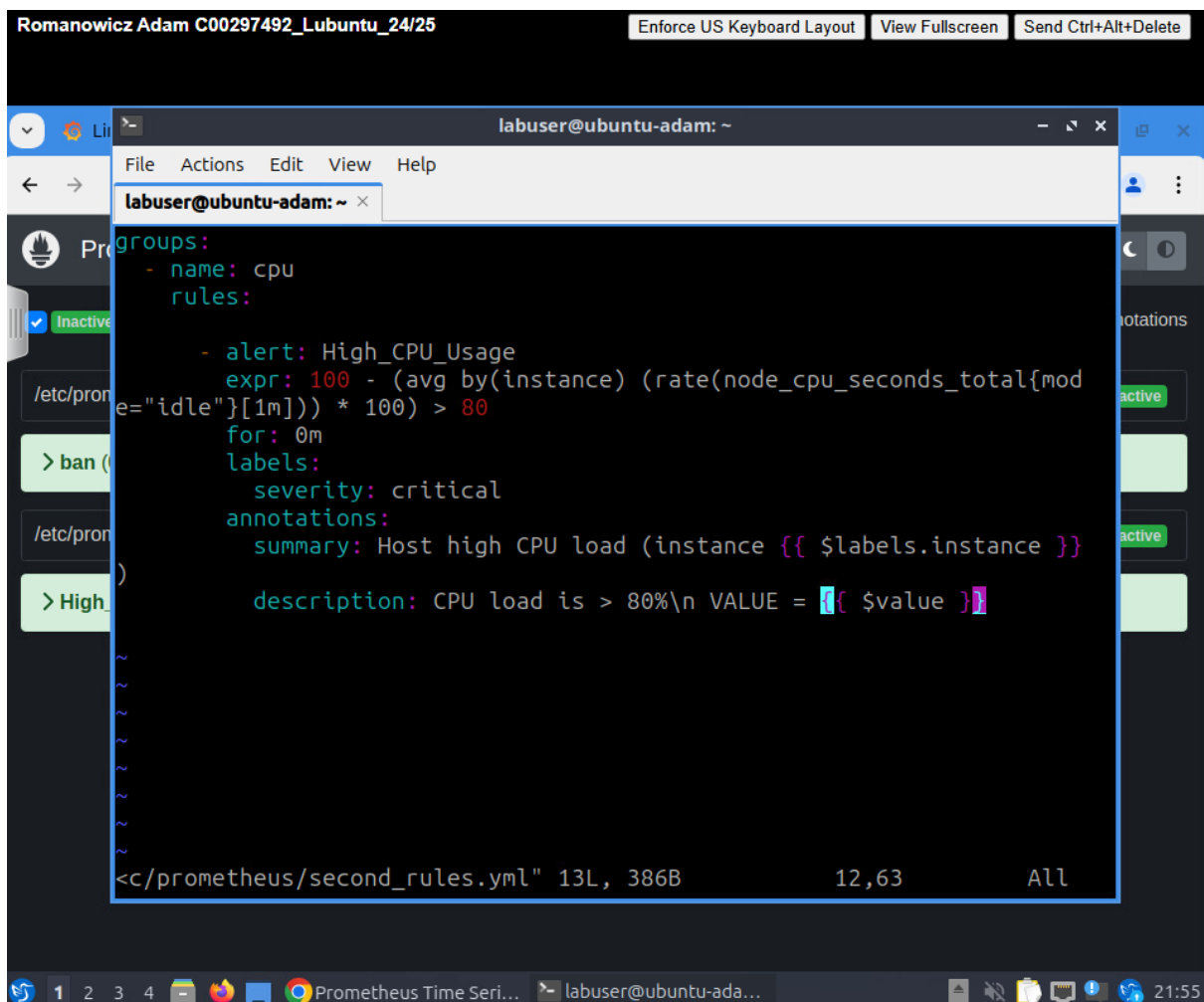
The screenshot shows a code editor window titled 'labuser@ubuntu-adam: ~' with a menu bar (File, Actions, Edit, View, Help). The editor displays the content of the file '/etc/prometheus/alert\_rules.yml'. The configuration is as follows:

```
groups:
- name: alerts
  rules:
  - alert: ban
    expr: f2b_jail_banned_current{instance=~"localhost:9191"} > 0
    for: 1m
    labels:
      severity: critical
    annotations:
      summary: Ip has been banned from {{ $labels.instance }}
      description: An ip has been banned for 3 consecutive failed log
```

At the bottom of the editor, a status bar indicates the file path '<tc/prometheus/alert\_rules.yml"', the size '12L, 333B', the cursor position '12,73', and the encoding 'All'.

Etc/Prometheus/alert\_rules.yml

This alert checks if the CPU average usage is above 80% for 1 minute and fires immediately when the criteria is met.



The screenshot shows a terminal window titled "labuser@ubuntu-adam: ~" with a menu bar (File, Actions, Edit, View, Help). The terminal displays the configuration for a Prometheus alert rule named "High\_CPU\_Usage". The configuration is as follows:

```
groups:
- name: cpu
  rules:
    - alert: High_CPU_Usage
      expr: 100 - (avg by(instance) (rate(node_cpu_seconds_total{mode="idle"}[1m])) * 100) > 80
      for: 0m
      labels:
        severity: critical
      annotations:
        summary: Host high CPU load (instance {{ $labels.instance }})
        description: CPU load is > 80%\n VALUE = {{ $value }}
```

The terminal status bar at the bottom shows the file path "c/prometheus/second\_rules.yml", file size "13L, 386B", line count "12,63", and "All" files selected. The system clock in the bottom right corner shows "21:55".

Etc/Prometheus/second\_rules.yml

Romanowicz Adam C00297492\_Lubuntu\_24/25

Enforce US Keyboard Layout View Fullscreen Send Ctrl+Alt

Linux Exporter Node - D Prometheus Time Series Alertmanager

localhost:9093/#/alerts

Alertmanager Alerts Silences

Filter Group

Custom matcher, e.g. env="production"

Expand all groups

webhook alertname="ban" +

2025-04-05T15:28:43.750Z + Info

instance="localhost:9191" + jail="ss"

```

# my global config
global:
  scrape_interval: 15s # Set the scrape interval to every 15 seconds.
  Default is every 1 minute.
  evaluation_interval: 15s # Evaluate rules every 15 seconds. The default
  is every 1 minute.
  # scrape_timeout is set to the global default (10s).

# Alertmanager configuration
alerting:
  alertmanagers:
    - static_configs:
      - targets:
        - localhost:9093

# Load rules once and periodically evaluate them according to the global
# 'evaluation_interval'.
rule_files:
  - "alert_rules.yml"
  - "second_rules.yml"

# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
scrape_configs:
  - job_name: 'prometheus'
    static_configs:
      - targets: ['localhost:9090']

```

<tc/prometheus/prometheus.yml" 57L, 1626B 12,19 Top

Etc/Prometheus/Prometheus.yml

This links alertmanager alerts to the slack channel via webhook and formats the the alert showing its status, alertname, instance,description and severity.

Romanowicz Adam C00297492\_Lubuntu\_24/25

Enforce US Keyboard Layout View Fullscreen Send Ctrl+Alt+Delete

labuser@ubuntu-adam: ~

```

global:
  resolve_timeout: 1m
  slack_api_url: 'https://hooks.slack.com/services/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'

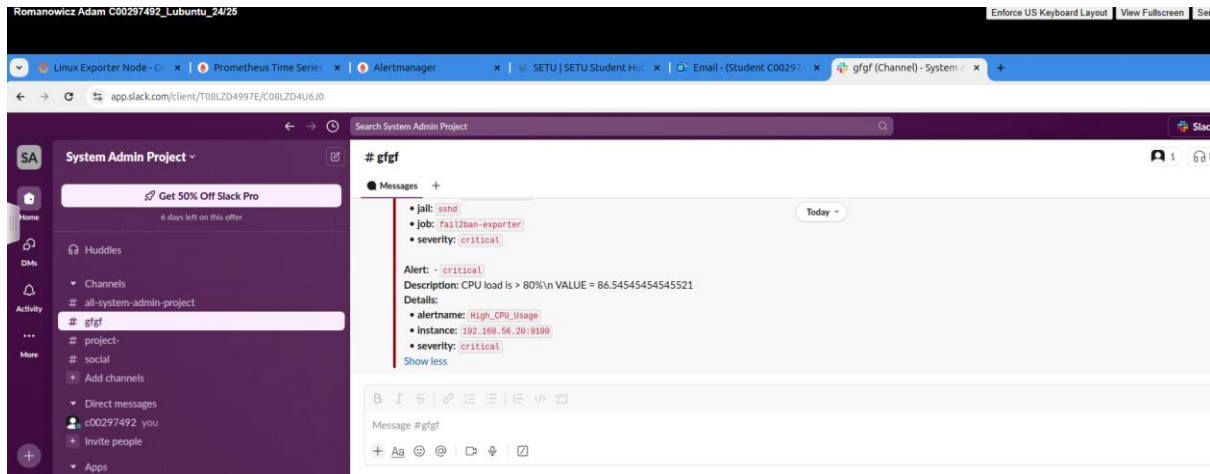
route:
  receiver: 'slack-notifications'

receivers:
- name: 'slack-notifications'
  slack_configs:
    - channel: '#gfg'
      send_resolved: true
      icon_url: https://avatars3.githubusercontent.com/u/3380462
      title: |-
        {{ range .Alerts.Firing }}{{ if eq .Status "Firing" }}{{ .Alerts.Firing | len }}{{ end }}{{ .CommonLabels.alertname }} for {{ .CommonLabels.job }}
        {{ if gt (len .CommonLabels) (len .GroupLabels) -}}
        {{ range .CommonLabels.Remove .GroupLabels.Names }}
        {{ range $index, $label := .SortedPairs -}}
        {{ if $index }}{{ end }}
        {{ $label.Name }}={{ $label.Value -}}
        {{ end -}}
        {{ end -}}
        {{ end -}}
      text: >-
        {{ range .Alerts -}}
        *Alert:* {{ .Annotations.title }}{{ if .Labels.severity }} - '{{ .Labels.severity }}'{{ end }}
        *Description:* {{ .Annotations.description }}
        *Details:*
        {{ range .Labels.SortedPairs }} *({{ .Name }}):* '{{ .Value }}'
        {{ end }}
      {{ end }}

```

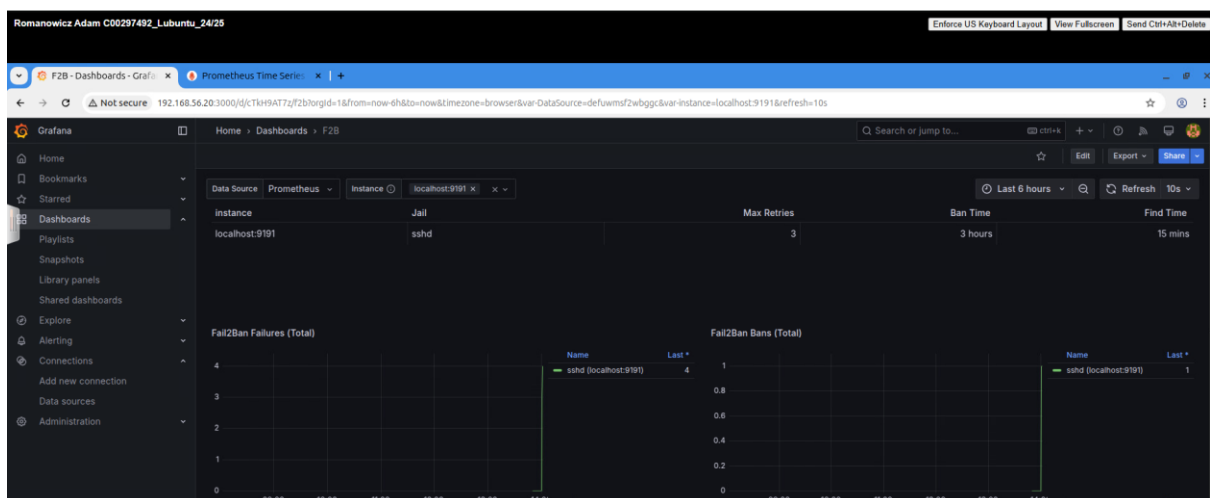
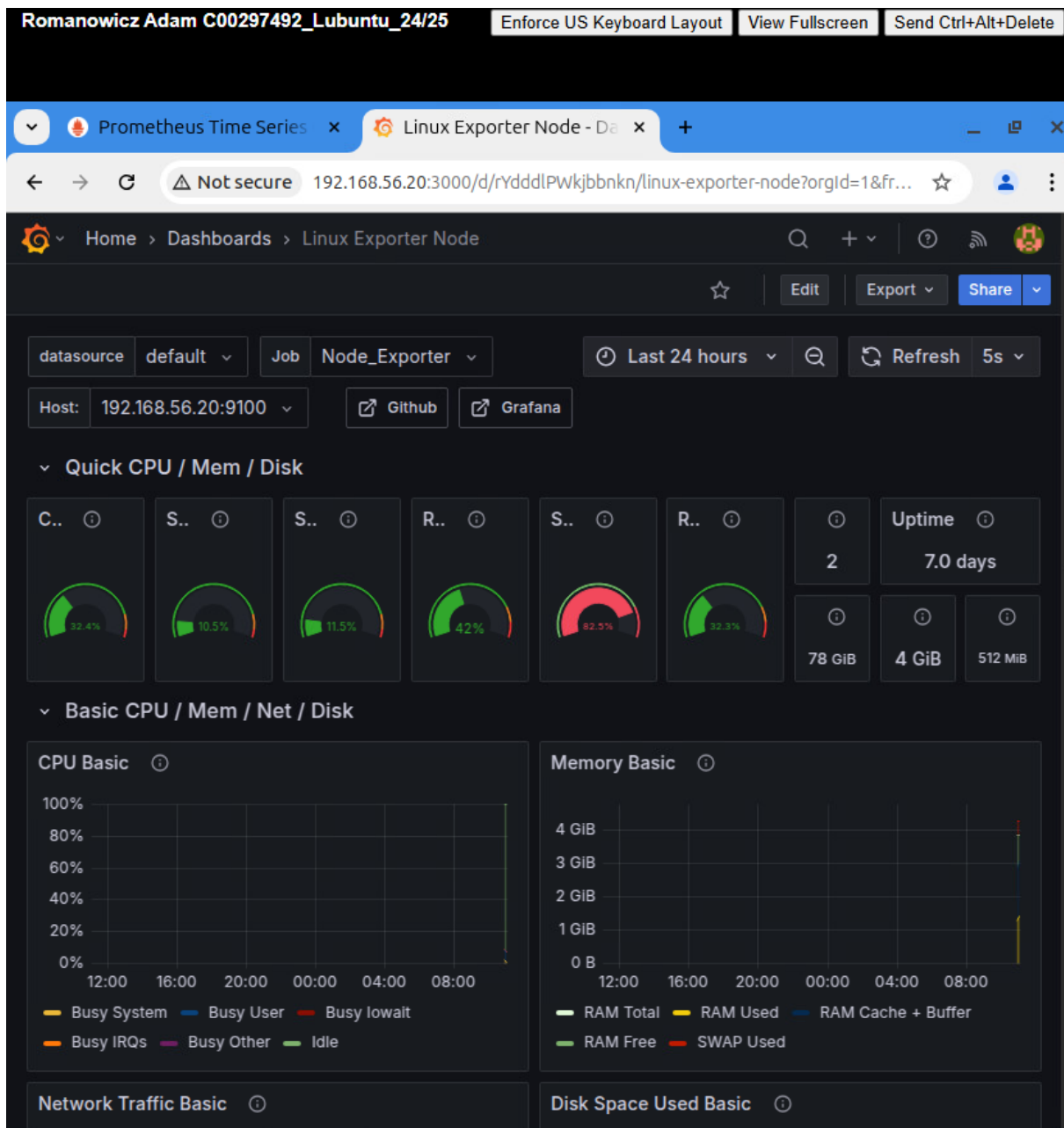
1,0-1 Top

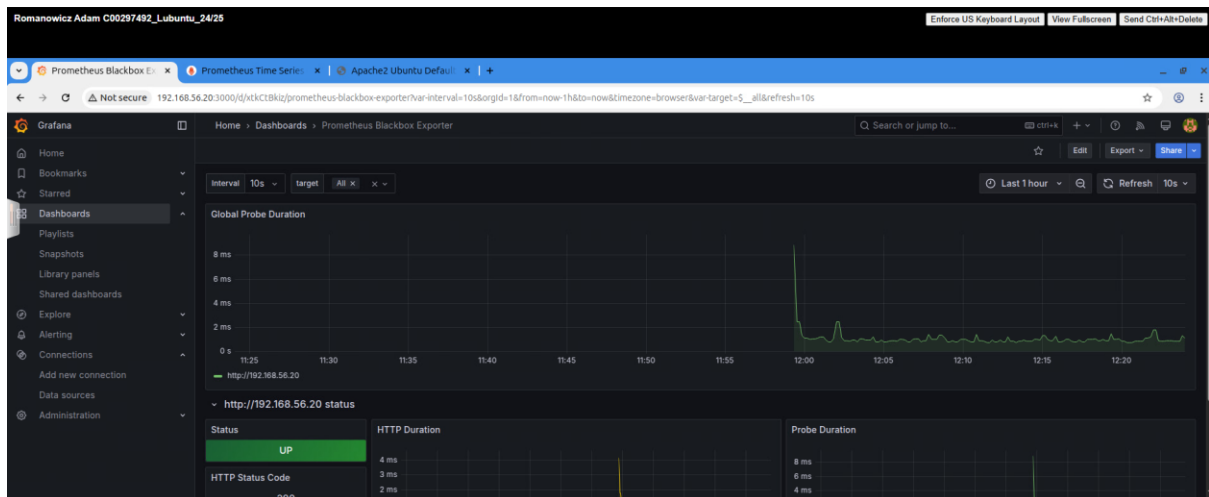
36L, 1162B 36,14 Bot



## Grafana

Grafana is used with Prometheus to provide a GUI and display metrics exposed by Prometheus and its exporters. This provides real-time insights to how the server is performing and its health. The three dashboards that are included are displayed below which show system resource usage, failed login attempts and blocked ips, and http traffic. To setup Grafana wget it off their official website then start and enable the service with systemctl. Enter the gui by typing <http://localhost:3000> and login using the username admin and password admin. Add Prometheus as a datasource by going into the gui and pressing data sources and adding the server ip and port (<http://localhost:9090/>). Create dashboards by pressing the + icon then select import which you can then import a json file or via ID.

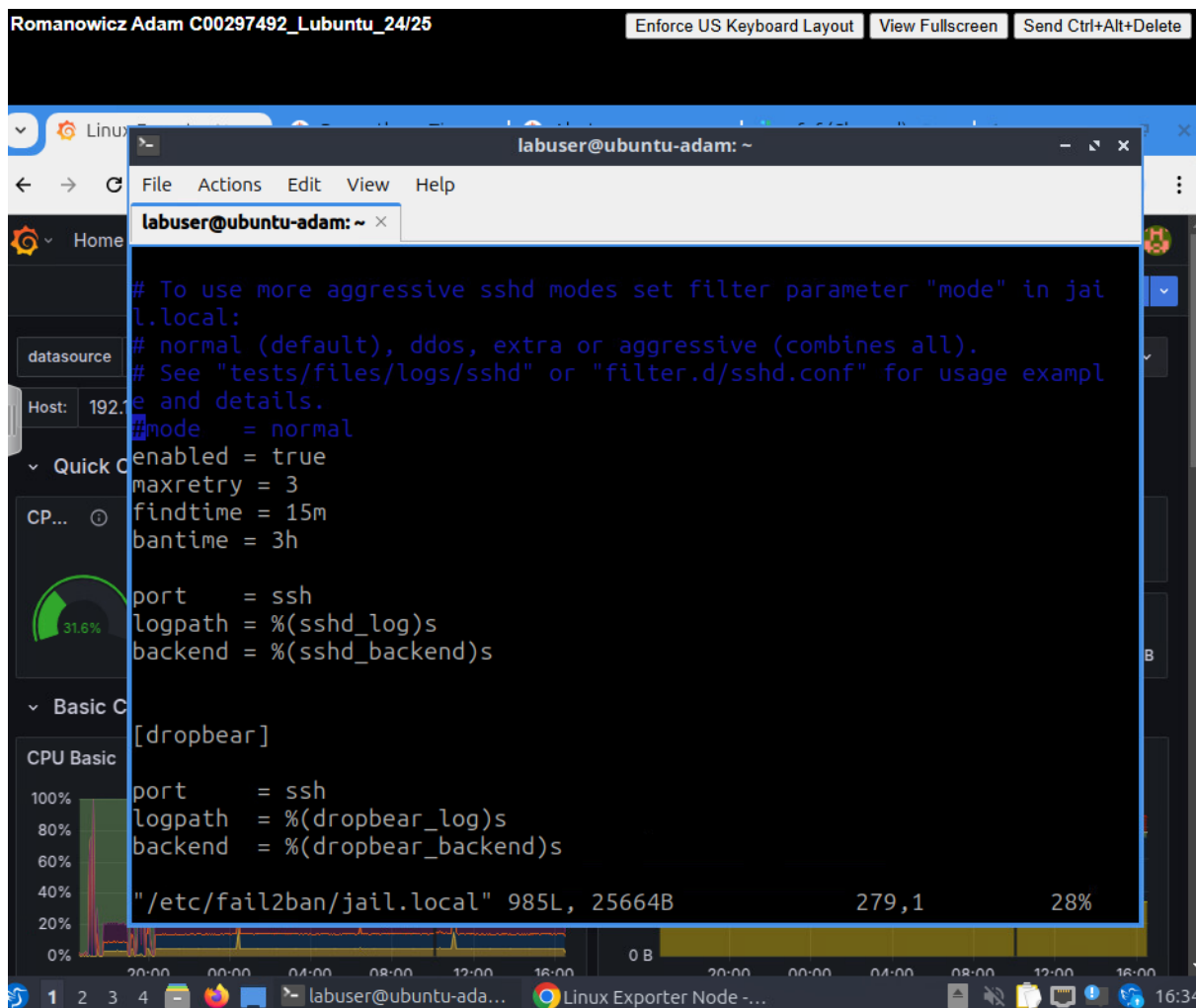




## Fail2ban

Fail2ban was configured on the server to protect it against brute force attacks and block any unauthorized agents from gaining access. It is currently configured to ban ips after 3 unsuccessful login attempts within a 15 minute time frame and bans them for 3 hours. To setup fail2ban install it using apt. Copy the /etc/fail2ban/jail.conf and name it jail.local. Edit this configuration file and uncomment the ssh section and change its settings to your specifications.





## Testing

To test the alerts try ssh into the ubuntu vm via the rocky vm and put in the wrong password 3 times. This will ban that ip for 3 hours and the ban alert will start firing. To test the High CPU usage alert type in the command “stress –cpu 2 –timeout 300”, this runs a stress test on the cpu and makes its average usage higher than 80% for 300 seconds which triggers the alert.

## **References**

[Install Prometheus and Grafana on Ubuntu 24.04 LTS](#)

<https://www.tecmint.com/install-fail2ban-ubuntu-24-04/>

<https://www.opsramp.com/guides/prometheus-monitoring/prometheus-blackbox-exporter/>

<https://www.geeksforgeeks.org/prometheus-blackbox-exporter/>

<https://github.com/hctrdev/fail2ban-prometheus-exporter?tab=readme-ov-file>

<https://developer.couchbase.com/tutorial-configure-alertmanager>

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