

# Systems Infrastructure and Security SEM2 Project

By: Jamie Dempsey

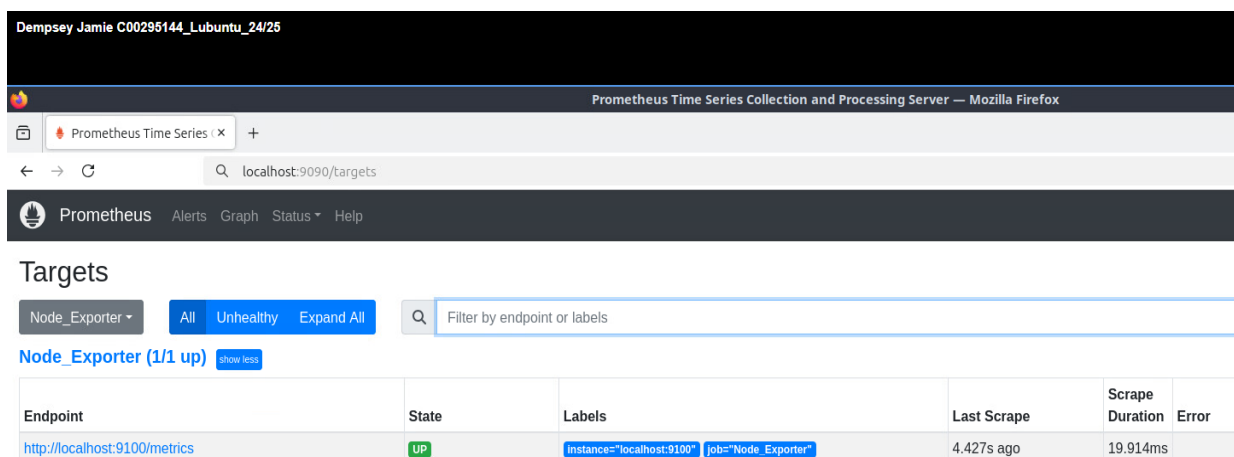
Student No: C00295144

For my SEM2 project in Systems Infrastructure & Security, I've been tasked with setting up an automated protection and monitoring environment. I'll be working with a Rocky Linux and Ubuntu server, using Prometheus to monitor the system's performance and trigger alerts when there are issues. I'll also be using **Grafana** to build dashboards, making it easier to visualize what's happening in real-time. To help maintain a secure environment, I'll be setting up **Fail2ban**, which can automatically block suspicious IP addresses based on failed login attempts. This will help me understand how to collect and analyze security data, detect potential threats, and respond to them quickly.

## Requirement 1: Prometheus Configuration & Alerts

To meet the first requirement of the project, I configured Prometheus on the Ubuntu server to scrape metrics from three main exporters (extras were added for demonstration purposes). Each exporter was configured inside the **prometheus.yml** file to offer a successful connection with Prometheus.

**Node Exporter** was installed to monitor metrics like CPU usage, memory usage and disk I/O. A job variable was created inside prometheus.yml to scrape from each server. I verified this was set up correctly by checking the targets section of the Prometheus site and making sure it appeared as **UP**.



Dempsey Jamie C00295144\_Lubuntu\_24/25

Prometheus Time Series Collection and Processing Server — Mozilla Firefox

Prometheus Time Series x +

localhost:9090/targets

Prometheus Alerts Graph Status Help

### Targets

Node\_Exporter All Unhealthy Expand All

Filter by endpoint or labels

Node\_Exporter (1/1 up) show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
<a href="http://localhost:9100/metrics">http://localhost:9100/metrics</a>	UP	instance="localhost:9100" job="Node_Exporter"	4.427s ago	19.914ms	

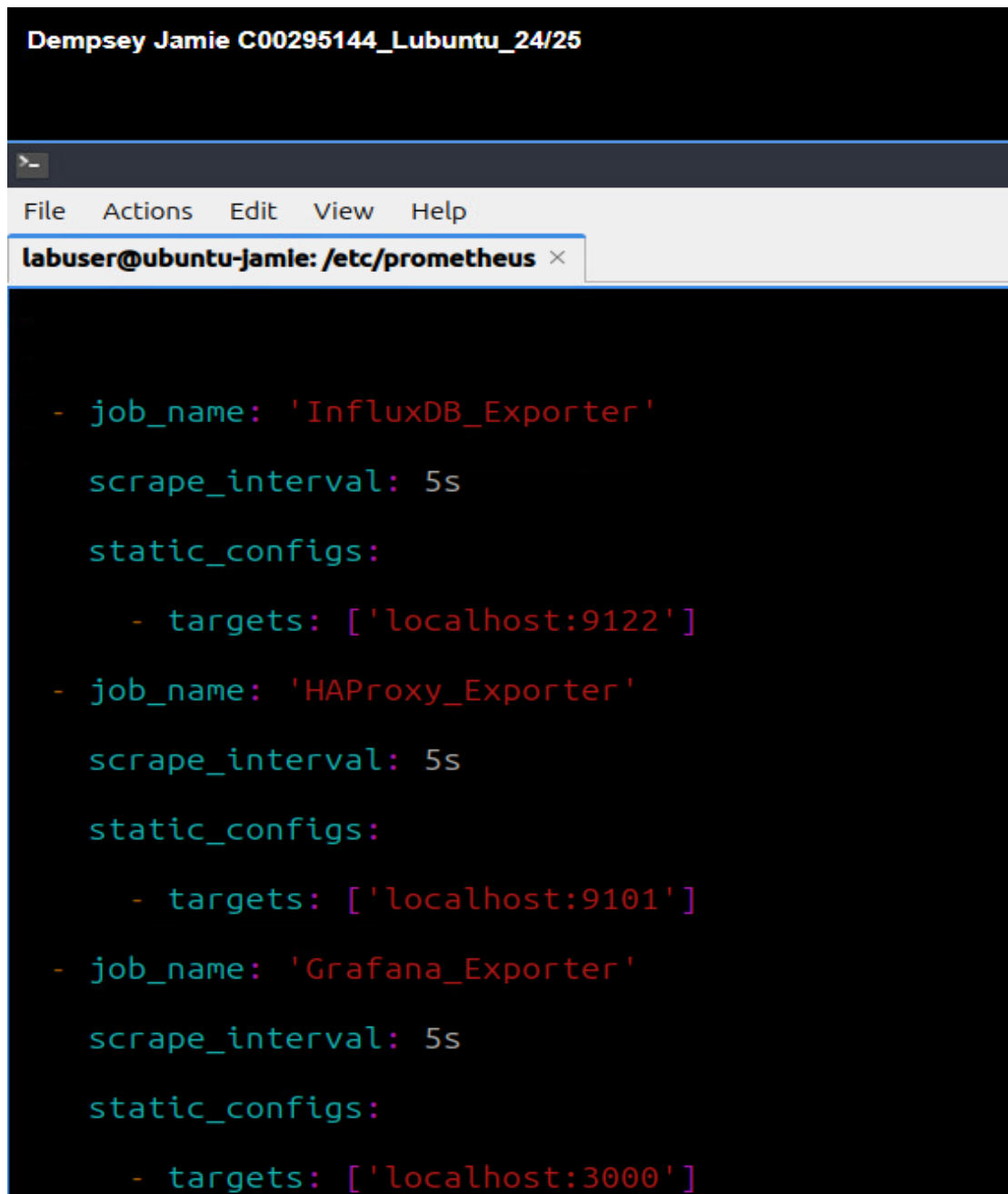
**Fail2ban Exporter** was installed to collect data on any failed login attempts through SSH. This exporter was also assigned a job variable to be able to scrape from the server. To test the functionality of the exporter, I simulated a failed SSH login attempt to confirm the metrics were updating accordingly.

**Blackbox Exporter** was installed to be able to monitor any external or internal endpoints that use HTTP, HTTPS, DNS, TCP or ICMP. A job variable was also assigned to this exporter, but with a slightly different approach. The job was configured to probe both local and remote services. This meant that the exporter could scrape from **localhost** and **https://example.com** sites.

```
Dempsey Jamie C00295144_Lubuntu_24/25

labuser@ubuntu-jamie: /etc/prometheus x
scrape_configs:
  - job_name: 'Node_Exporter'
    scrape_interval: 5s
    static_configs:
      - targets: ['localhost:9100'] # 192.168.56.20:9100
  - job_name: 'Fail2ban_Exporter'
    scrape_interval: 5s
    static_configs:
      - targets: ['localhost:9192'] # 192.168.56.20:9192
  - job_name: 'Blackbox_Exporter'
    metrics_path: /probe
    params:
      module: [http_2xx]
    static_configs:
      - targets: ['localhost:9115'] # 192.168.56.20:9115
    relabel_configs:
      - source_labels: [__address__]
        target_label: __param_target
      - source_labels: [__param_target]
        target_label: instance
      - target_label: __address__
        replacement: localhost:9115
```

Other exporters configured into Prometheus for demonstration purposes.



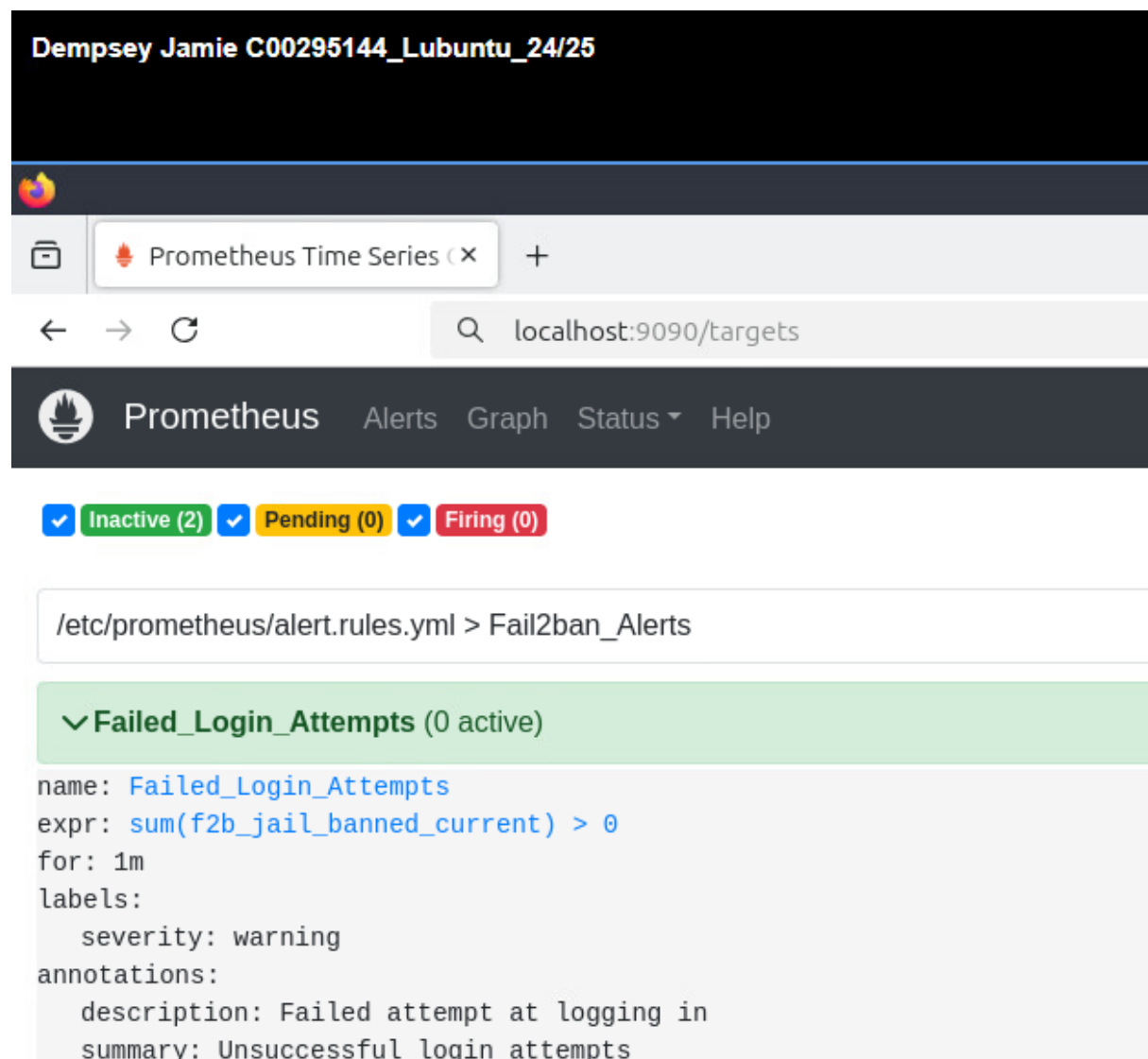
```
Dempsey Jamie C00295144_Lubuntu_24/25

labuser@ubuntu-jamie: /etc/prometheus

- job_name: 'InfluxDB_Exporter'
  scrape_interval: 5s
  static_configs:
    - targets: ['localhost:9122']
- job_name: 'HAProxy_Exporter'
  scrape_interval: 5s
  static_configs:
    - targets: ['localhost:9101']
- job_name: 'Grafana_Exporter'
  scrape_interval: 5s
  static_configs:
    - targets: ['localhost:3000']
```

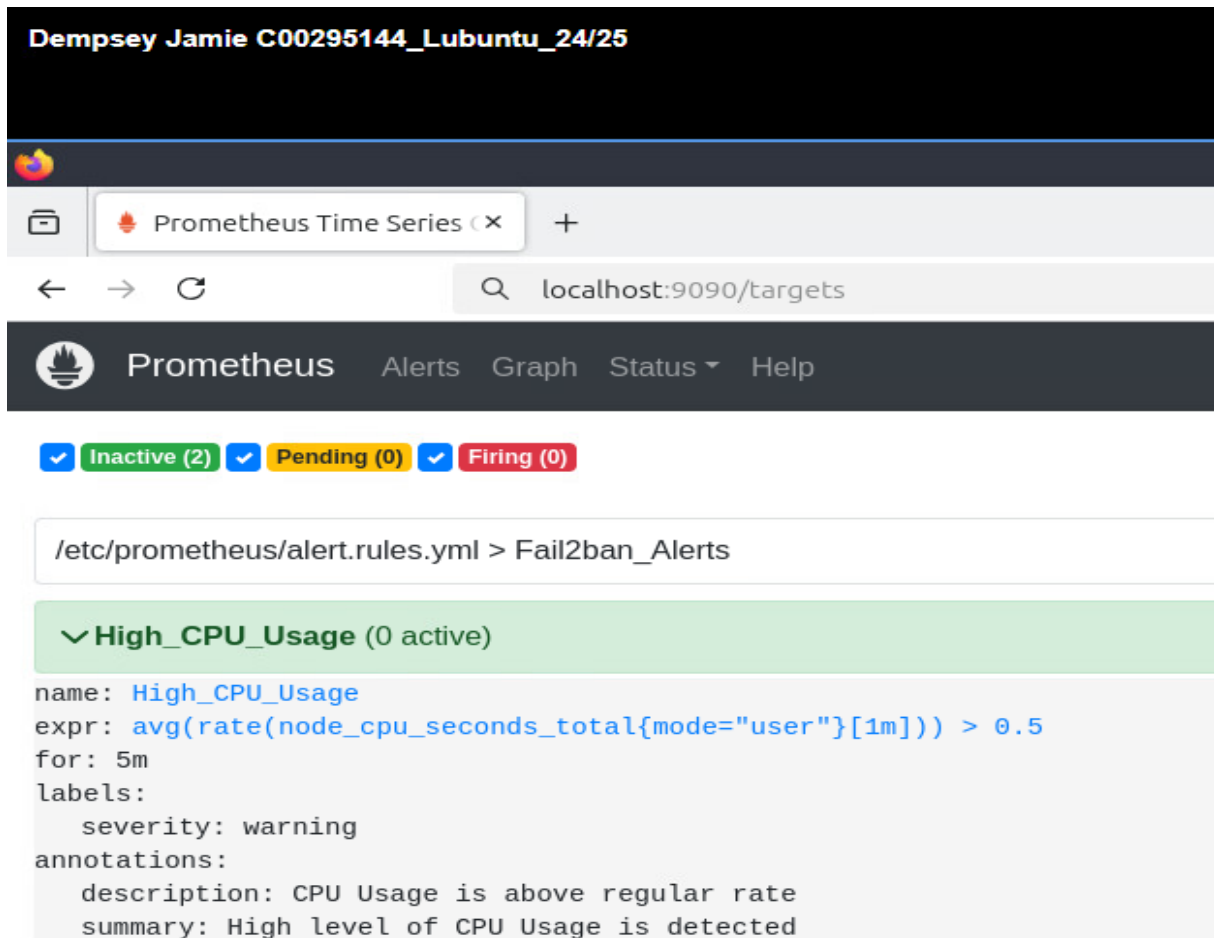
To enable alerts on the system, I downloaded the Prometheus Alertmanager package to notify me about potential security and performance issues. These alerts were configured in the **alert\_rules.yml** file as well as referenced inside **prometheus.yml** to connect them for accurate execution and usage.

**Failed Login Attempts** through SSH was the first rule that I incorporated. This alert uses metrics from the **Fail2ban Exporter**. The alert executes once a failed login attempt has been identified through the SSH jail system. This is useful for monitoring any unusual login activity. I tested this alert by going into my Rocky Linux server and attempting to SSH into the Ubuntu server with fake credentials. This confirmed that the alert executed accordingly.



**High CPU Usage** detection was my second alert. As stated in the name, this alert monitors the CPU load using metrics from the **Node Exporter**. The alert will execute if CPU usage fluctuates above 50% for more than a minute. This is a useful alert for spotting any performance issues or processes that might be putting the CPU under too much pressure. I tested this by running CPU-heavy material to trigger the alert to ensure the functionality was correct.

Dempsey Jamie C00295144\_Lubuntu\_24/25



The image shows the Prometheus web interface. At the top, there's a header with the Prometheus logo and navigation links: Alerts, Graph, Status, and Help. Below the header, there's a status bar showing 'Inactive (2)', 'Pending (0)', and 'Firing (0)' alerts. The main content area displays the configuration for the 'Fail2ban\_Alerts' rule file. A section titled 'High\_CPU\_Usage (0 active)' is expanded, showing the following configuration:

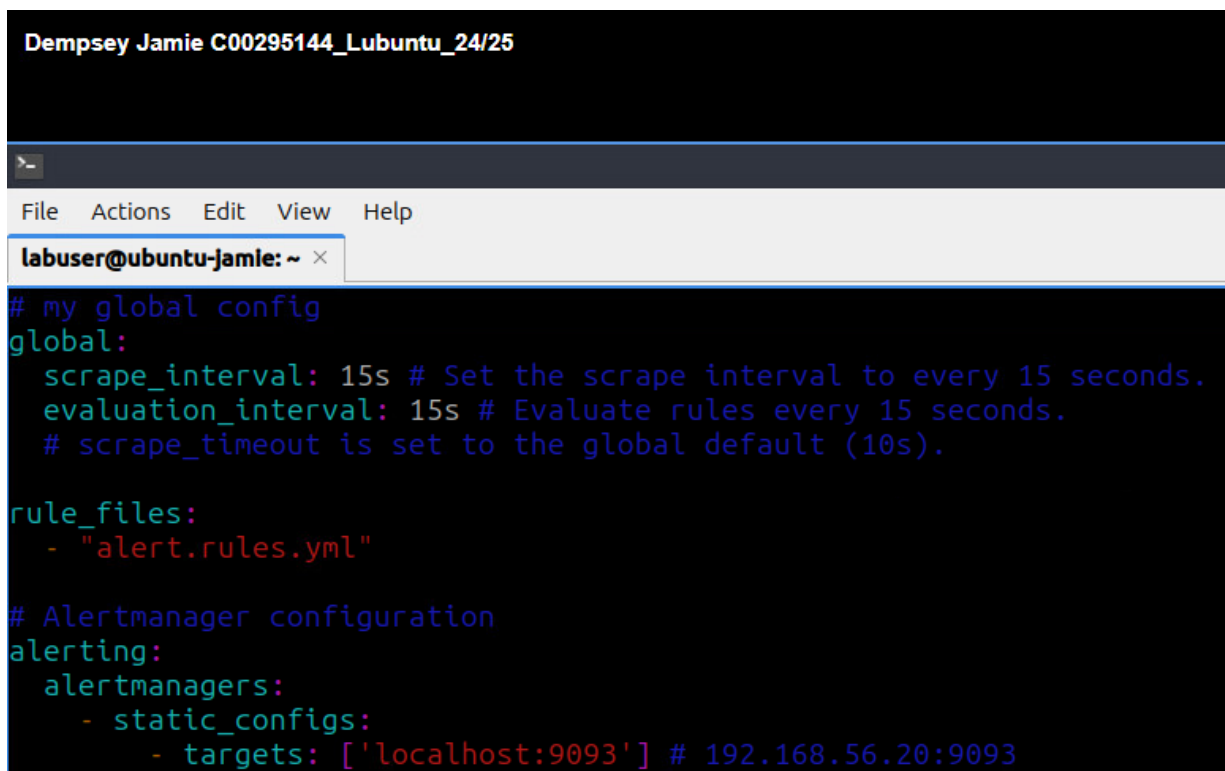
```

name: High_CPU_Usage
expr: avg(rate(node_cpu_seconds_total{mode="user"}[1m])) > 0.5
for: 5m
labels:
  severity: warning
annotations:
  description: CPU Usage is above regular rate
  summary: High level of CPU Usage is detected

```

Rules were defined in **alert\_rules.yml** and referenced in **prometheus.yml**

Dempsey Jamie C00295144\_Lubuntu\_24/25



The image shows a terminal window with the following content:

```

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

rule_files:
  - "alert.rules.yml"

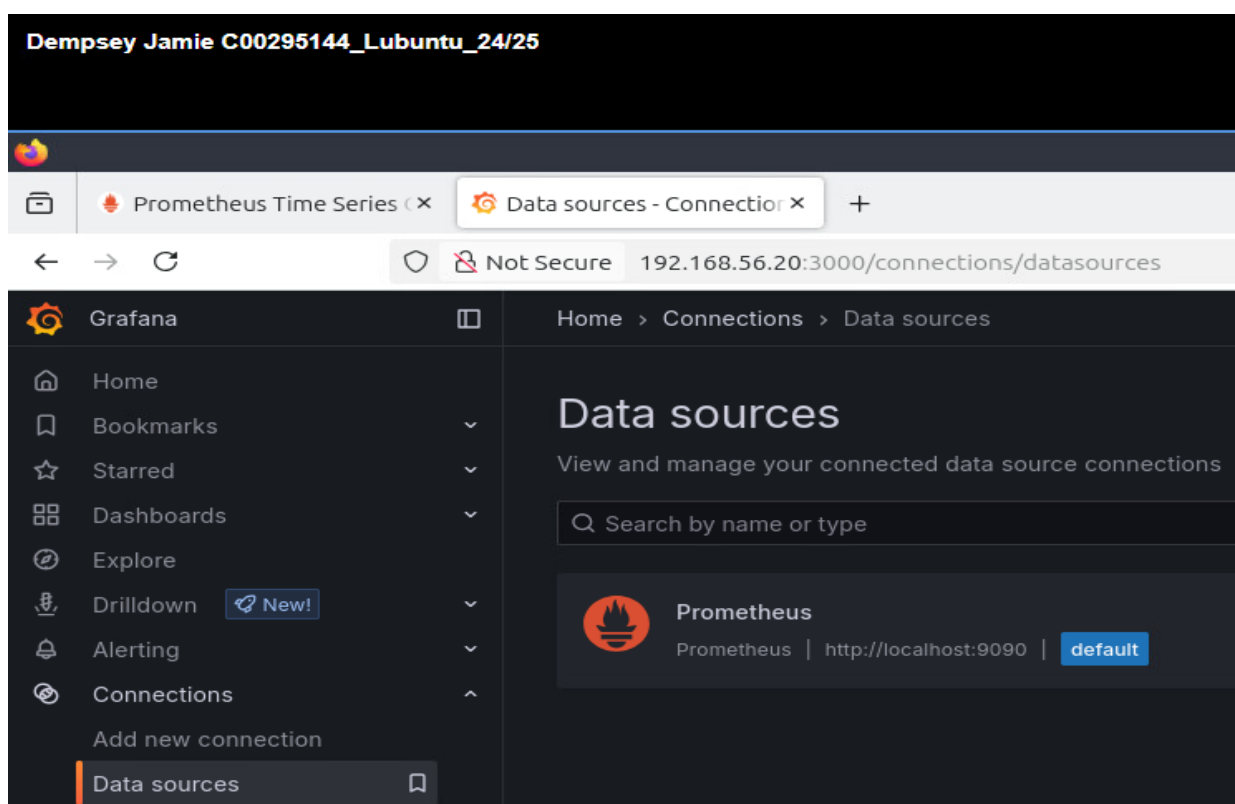
# Alertmanager configuration
alerting:
  alertmanagers:
    - static_configs:
      - targets: ['localhost:9093'] # 192.168.56.20:9093

```

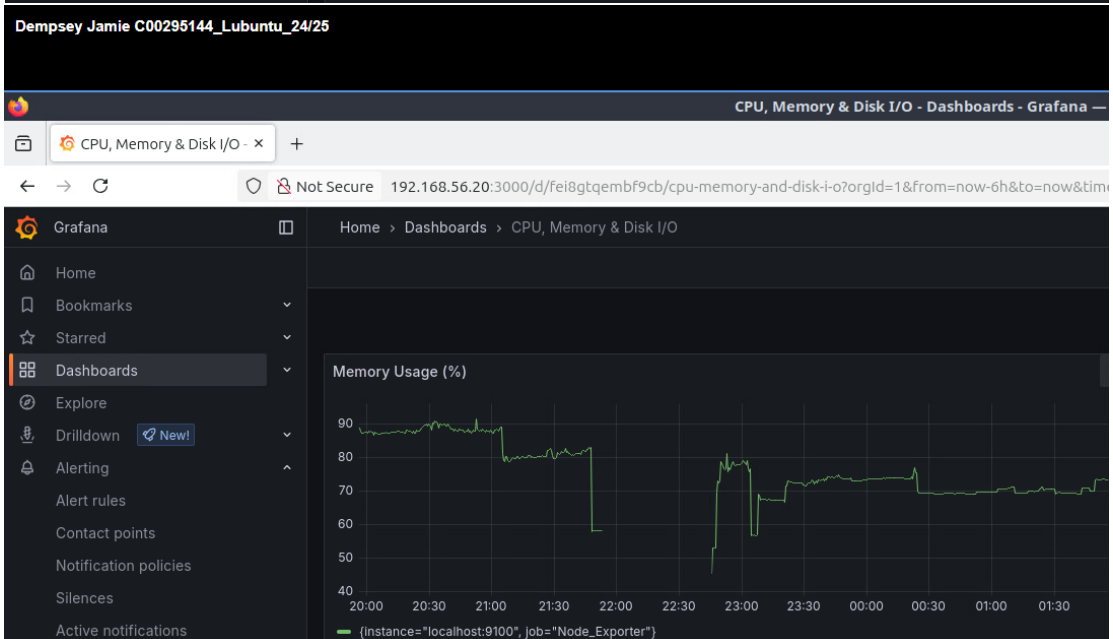
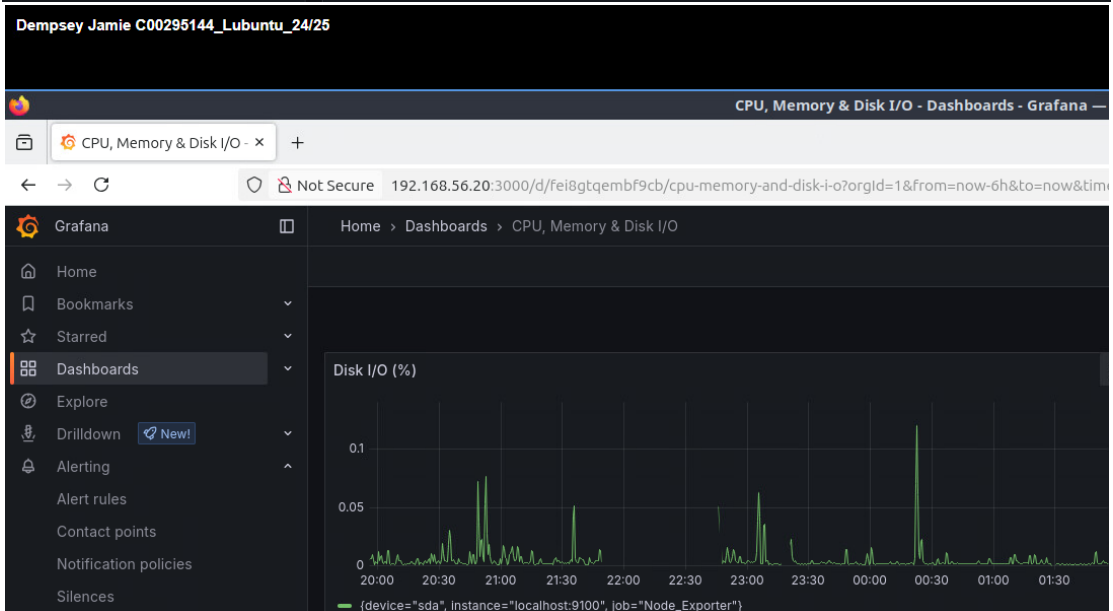
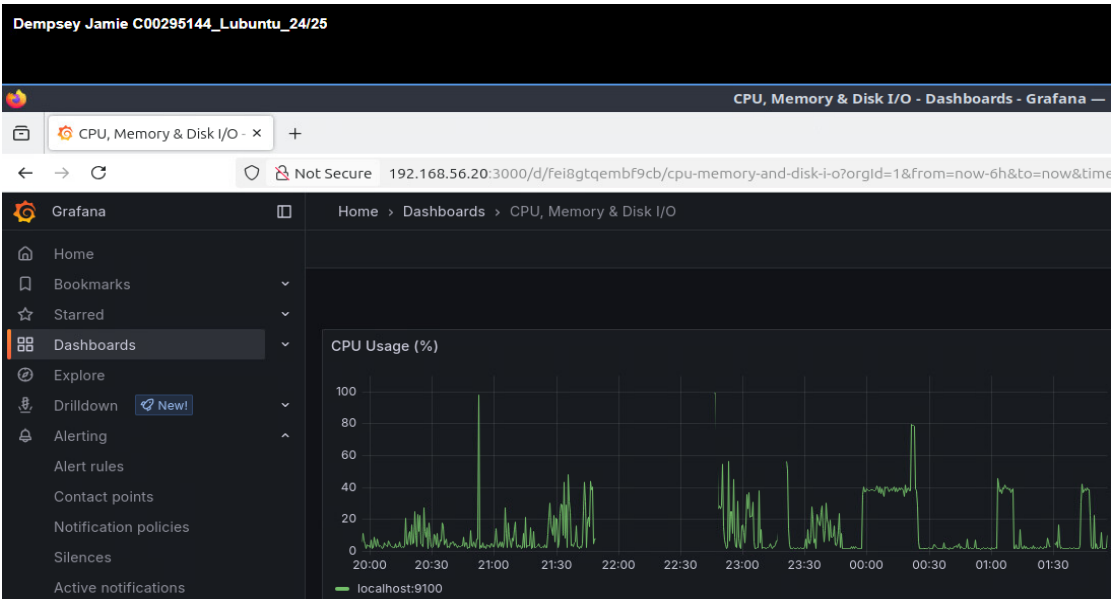
## Requirement 2: Grafana Configuration & Dashboards

For this part of the project, I configured Grafana to be able to connect to the Prometheus server as a data source. This allowed me to create dashboards that provide me with real-time insights into the performance of the system.

To connect Grafana to Prometheus, I began by adding Prometheus as a data source on the Grafana website. This involved me navigating to the **Data Sources** section, selecting **Prometheus** and then entering the Prometheus server URL to allow for a secure connection. Grafana could then pull data.



My first dashboard on Grafana displays typical **System Resource Data**, which is the CPU usage, memory consumption and disk I/O. I used the **Prometheus PromQL query** to create individual panels for each resource. For CPU, it displays the percentage of CPU usage over time, for memory, it shows total memory consumption and for disk I/O, I used a graph to display any disk usage trends. This dashboard helps me monitor the overall health of the system in a real-time environment with an easily accessible interface.

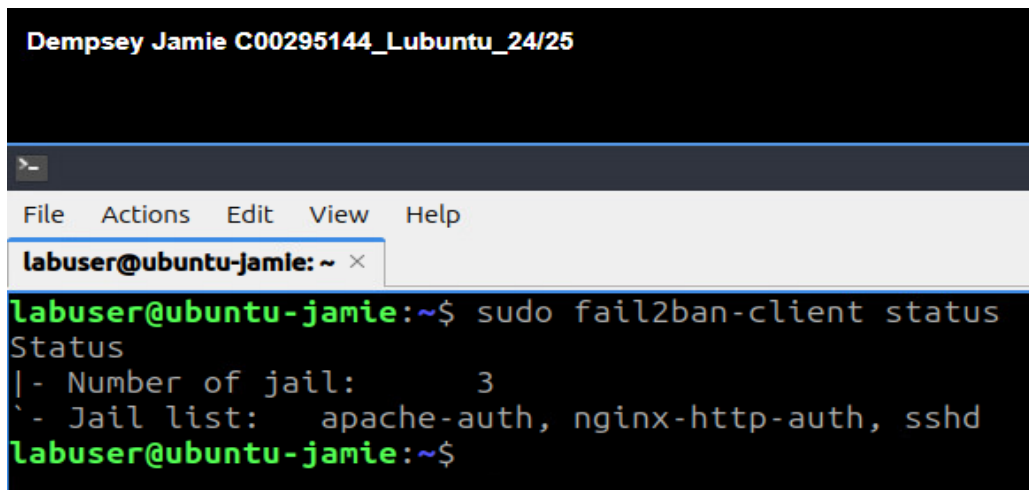


### Requirement 3: Fail2ban Integration & SSH

To complete the final requirement of the project, I configured Fail2ban to monitor and protect SSH as well as web services that are frequently running.

Firstly, I had to install **OpenSSH** on Ubuntu so I could access an SSH login into Ubuntu from Rocky Linux. I then enabled the default **SSHD** jail in the fail2ban configuration to monitor the **auth.log** file for failed login attempts.

I enabled both jails for **apache-auth** and **nginx-http-auth** to monitor HTTP access, allowing for enhanced security, and importantly, the **sshd** jail to guarantee the functionality of SSH protection for users on the Ubuntu server.

A terminal window titled 'Dempsey Jamie C00295144\_Lubuntu\_24/25' showing a terminal session. The user 'labuser@ubuntu-jamie' runs the command 'sudo fail2ban-client status'. The output shows the status of the Fail2ban service, including the number of jails (3) and the list of jails (apache-auth, nginx-http-auth, sshd).

```
Dempsey Jamie C00295144_Lubuntu_24/25
labuser@ubuntu-jamie:~$ sudo fail2ban-client status
Status
|- Number of jail:      3
`- Jail list:  apache-auth, nginx-http-auth, sshd
labuser@ubuntu-jamie:~$
```

I set specific ban rules in the **jail.local** for SSH. The max retry, which is how many attempts you have to enter the correct credentials, is set at **3**, so once you surpass the 3 attempts, the credentials you entered will automatically be banned from SSH access for **5 minutes**, which has been set under the ban time section. The 5-minute ban was implemented for demonstration purposes only to allow for easy testing. This time frame should be increased to ensure maximum security. I tested this service by attempting to SSH from Rocky Linux into the Ubuntu server while using fake credentials to trigger the Fail2ban service. This helps block out unauthorized access to the server.



### Dempsey Jamie C00295144\_Lubuntu\_24/25

```
>_
File Actions Edit View Help
labuser@ubuntu-jamie: /etc/fail2ban x

[sshd]

enabled = true
port = ssh
filter = sshd
logpath = /var/log/auth.log
backend = %(sshd_backend)s
maxretry = 3
bantime = 5m
```

### Dempsey Jamie C00295144\_Lubuntu\_24/25

```
>_
File Actions Edit View Help
labuser@ubuntu-jamie: ~ x

labuser@ubuntu-jamie:~$ sudo fail2ban-client status sshd
Status for the jail: sshd
|- Filter
| |- Currently failed: 0
| |- Total failed: 6
| `-- File list: /var/log/auth.log
`- Actions
    |- Currently banned: 1
    |- Total banned: 2
    `-- Banned IP list: 192.168.56.10
labuser@ubuntu-jamie:~$
```

### Dempsey Jamie C00295144\_RockyLinux\_24/25

```
[labuser@rocky-jamie ~]# ssh user@192.168.56.20
user@192.168.56.20's password:
Permission denied, please try again.
user@192.168.56.20's password:
Permission denied, please try again.
user@192.168.56.20's password:
Connection closed by 192.168.56.20 port 22
[labuser@rocky-jamie ~]# ssh user@192.168.56.20
ssh: connect to host 192.168.56.20 port 22: Connection refused
[labuser@rocky-jamie ~]#
```