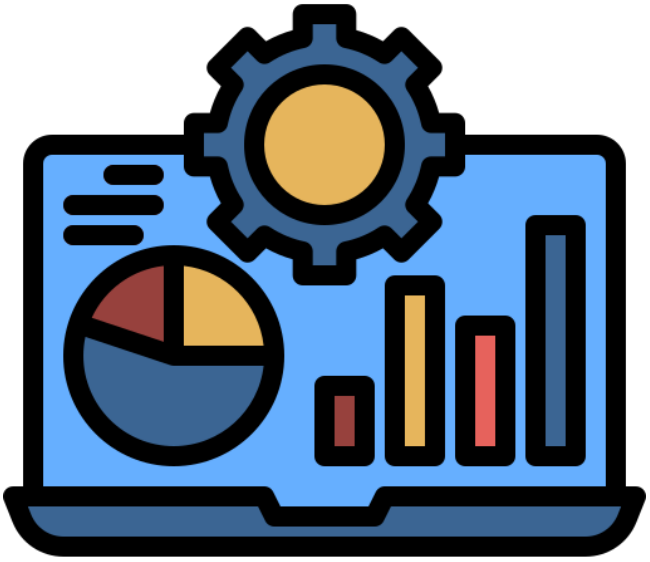

Infrastructure Monitoring with Prometheus and Grafana

Prepared By:
Mahmoud Alsalloum

Table of contents

Contents

Technology Stack

Ubuntu Server – Base OS hosting the monitoring stack and Monitored system running services like Nextcloud and Prometheus.

Prometheus – Collects and stores time-series metrics from the system.

Grafana – Visualizes metrics through customizable dashboards.

Node-exporter – Provides detailed system metrics such as CPU, memory, disk, and network usage.

Project Overview

Welcome to my project!

I'm honored that you are visiting my portfolio website and taking the time to explore my work. In this project, I implemented a monitoring solution on my Ubuntu server using Prometheus and Grafana to gain real-time insights into system performance and resource usage. The server already runs a self-hosted Nextcloud instance, and to ensure its reliability and smooth operation, I introduced a lightweight monitoring stack. With node-exporter feeding system metrics into Prometheus, Grafana visualizing them through interactive dashboards, I was able to track CPU load, memory usage, disk activity, and network throughput. This setup enables proactive issue detection, efficient resource management, and reflects core responsibilities in Site Reliability Engineering (SRE) – a vital aspect of the DevOps field. By observing trends and performance in real time, I can ensure that my infrastructure remains stable, secure, and optimized for future growth.

Implementation Steps

To make sure that my system is up to date I ran this command:

```
mahmoud@nextcloud:~$ sudo apt update && sudo apt full-upgrade -y
```

I download Prometheus directly from the official source and unpack it in a temporary folder. See Figure 1.

```
mahmoud@nextcloud:~$ cd /tmp
wget https://github.com/prometheus/prometheus/releases/download/v2.52.0/prometheus-2.52.0.linux-amd64.tar.gz
tar -xvf prometheus-2.52.0.linux-amd64.tar.gz
cd prometheus-2.52.0.linux-amd64
```

Figure 1: The installation of Prometheus

I place the Prometheus executable where they can be easily run and put configuration and console files in /etc/prometheus. The data directory is created at /var/lib/prometheus to store metrics data. See Figure 2.

```
mahmoud@nextcloud:/tmp/prometheus-2.52.0.linux-amd64$ sudo cp prometheus /usr/local/bin/
sudo cp promtool /usr/local/bin/
sudo mkdir -p /etc/prometheus /var/lib/prometheus
sudo cp -r consoles /etc/prometheus
sudo cp -r console_libraries /etc/prometheus
sudo cp prometheus.yml /etc/prometheus/
```

Figure 2: Moving binaries and config files to proper locations

I created a systemd service to manage Prometheus. sudo nano /etc/systemd/system/prometheus.service. This service file tells Ubuntu how to start and manage Prometheus in the background. It makes sure Prometheus will restart if it crashes and will start automatically on boot. See Figure 3.

```
  GNU nano 7.2                  /etc/systemd/system/prometheus.service *
[Unit]
Description=Prometheus Monitoring
After=network.target

[Service]
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

Restart=always

[Install]
WantedBy=multi-user.target
```

Figure 3: Add a configuration file for Prometheus

I reloaded systemd to recognize the new service, enabling it to run on startup, and start it now. Prometheus is now accessible at http://192.168.178.50:9090. See Figure 4.

```
mahmoud@nextcloud:/tmp/prometheus-2.52.0.linux-amd64$ sudo systemctl daemon-reexec
sudo systemctl enable prometheus
sudo systemctl start prometheus
```

Figure 4: Start and enable Prometheus

After successfully installing and starting Prometheus, I accessed the web interface through my browser at http://192.168.178.50:9090. The screenshot below shows the Prometheus UI,

where I can run queries, explore metrics, and verify that the monitoring service is working correctly. See Figure 5.

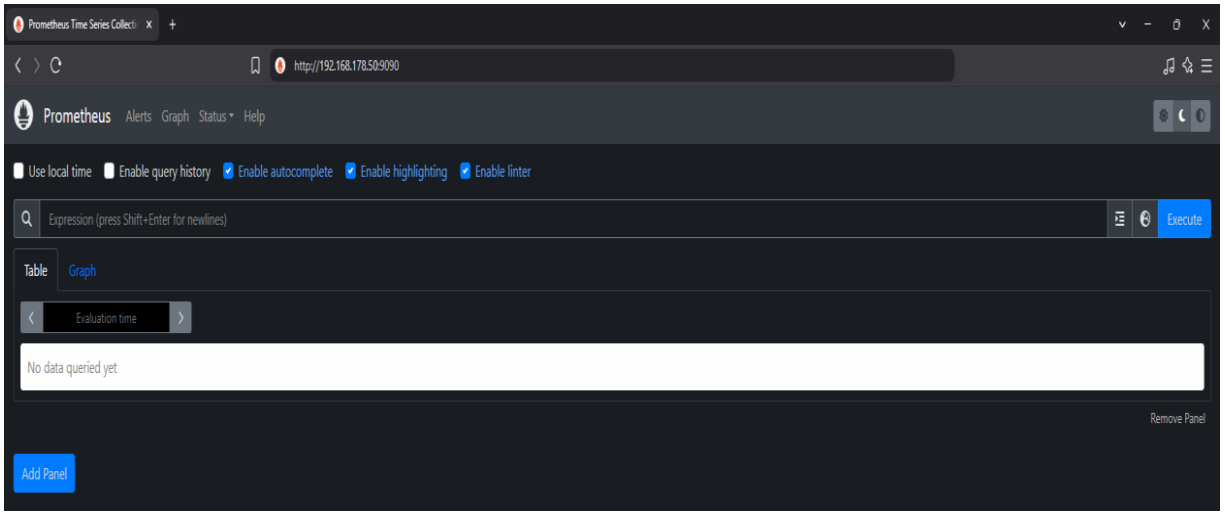

Figure 5: The web interface of Prometheus

Grafana is not included in Ubuntu's default repositories. I add the official Grafana APT repository and its GPG key so that the system can securely fetch and verify the Grafana packages. See Figure 6.

```
mahmoud@nextcloud:~$ sudo apt install -y software-properties-common
sudo add-apt-repository "deb [arch=amd64] https://packages.grafana.com/oss/deb stable main"
wget -q -O - https://packages.grafana.com/gpg.key | sudo apt-key add -
sudo apt update
```

Figure 6: The installation of Grafana

I start the Grafana service and configure it to launch automatically whenever the server boots up. See Figure 7.

```
mahmoud@nextcloud:~$ sudo systemctl enable grafana-server
sudo systemctl start grafana-server
```

Figure 7: Start and enable Grafana

I opened my browser and navigated to: http://192.168.178.50:3000 to access the Grafana web interface where I can connect data sources like Prometheus and create dashboards for visualizing metrics. See Figure 8.

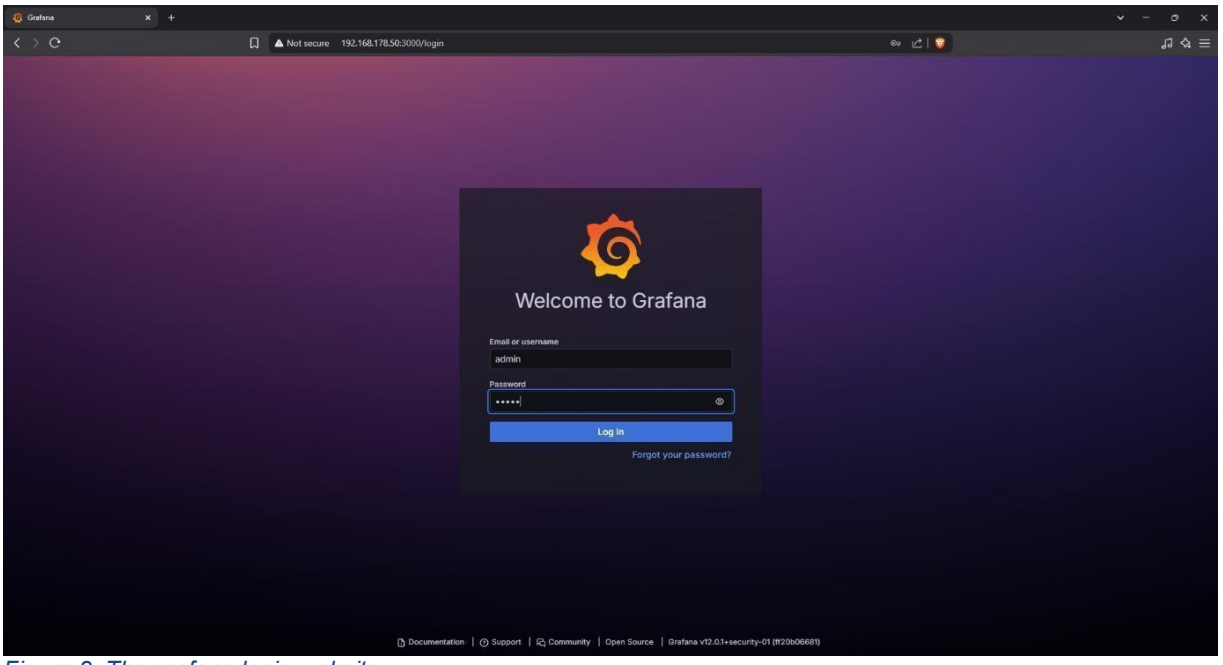

Figure 8: The grafana login website

I entered “admin” and “admin” for default username and default password before I was asked to set a new password on the first login. After I created a new password for my admin- account I entered the Grafana admin-console. In Grafana I clicked on “Data Sources” and chose “Add data source” Prometheus. I entered in the URL field: http://localhost:9090. See Figure 9.

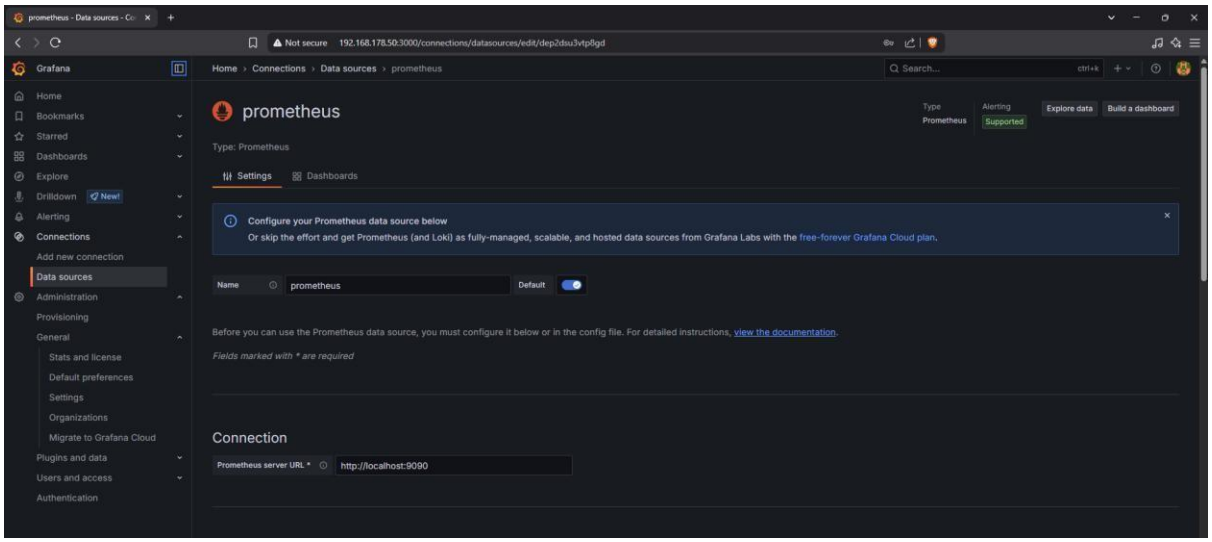

Figure 9: Set Prometheus server URL in Grafana

This step links Grafana with Prometheus, allowing it to pull in all metrics collected by Prometheus and use them in dashboards.

In the next step I imported the dashboard by clicking on the import dashboard and entering the value 1860. After that I selected the Prometheus data source as shown in Figure 10.

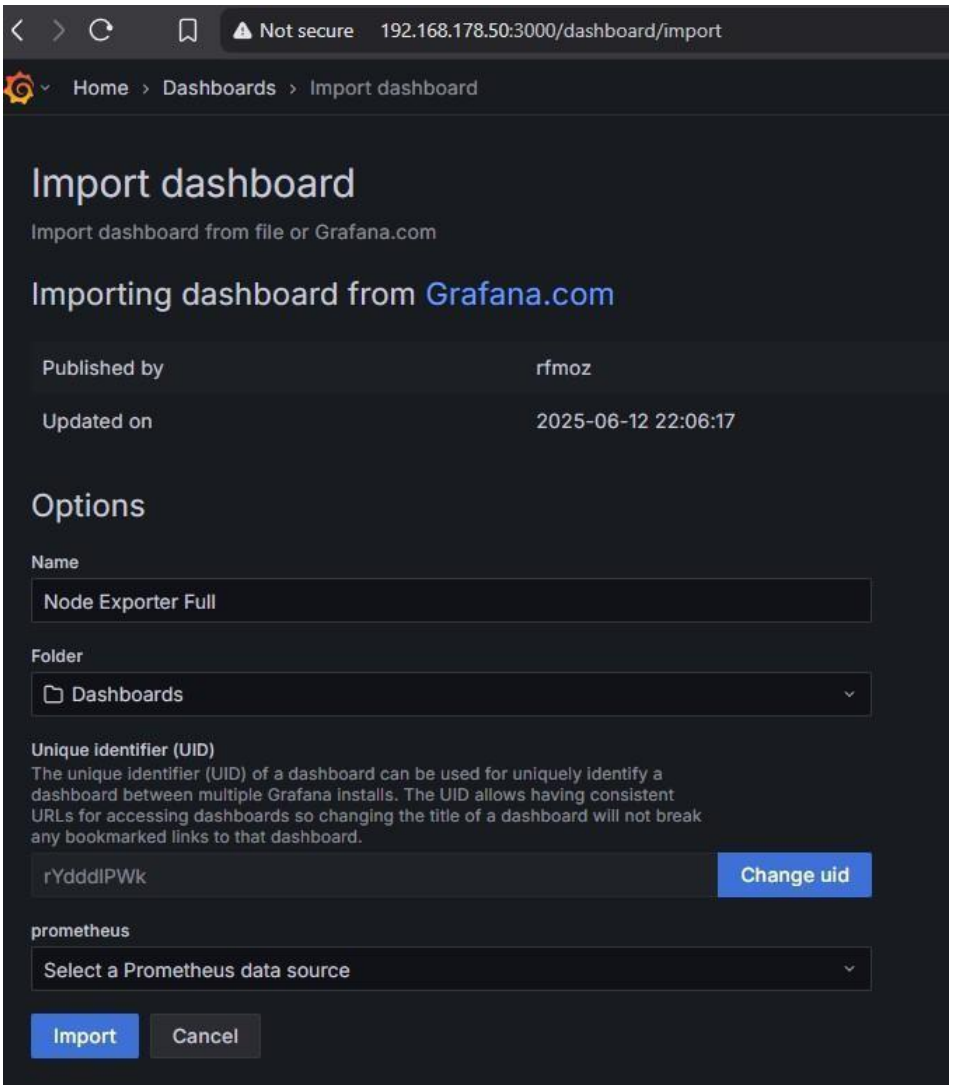

Figure 10: Import prometheus data source

After that the dashboard was opened but it was empty. No metrics were received from Prometheus. See Figure 11.

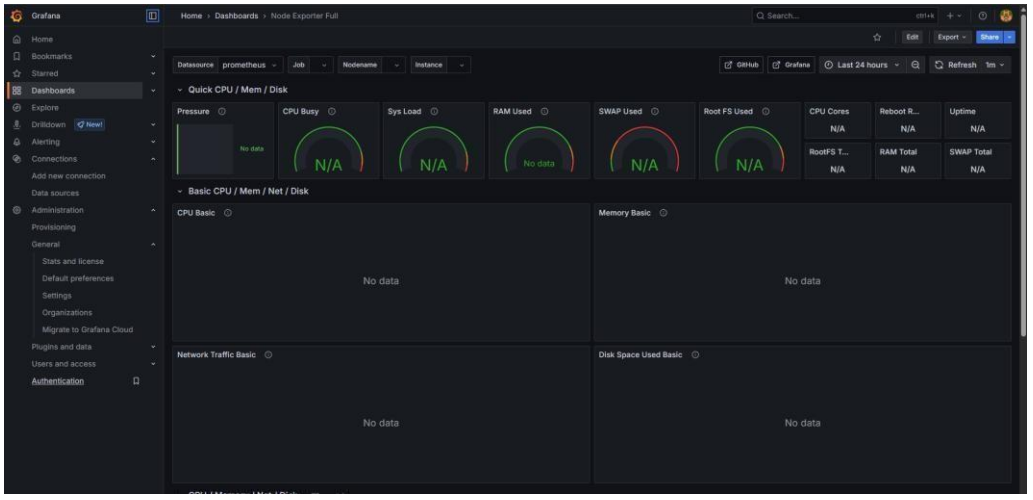

Figure 11: The Grafana empty dashboard

To resolve this failure, I opened the configuration file Prometheus.yml and I found out that node_exporter was not configured in the file. I edited the file and added the scrape config. See Figure 12.

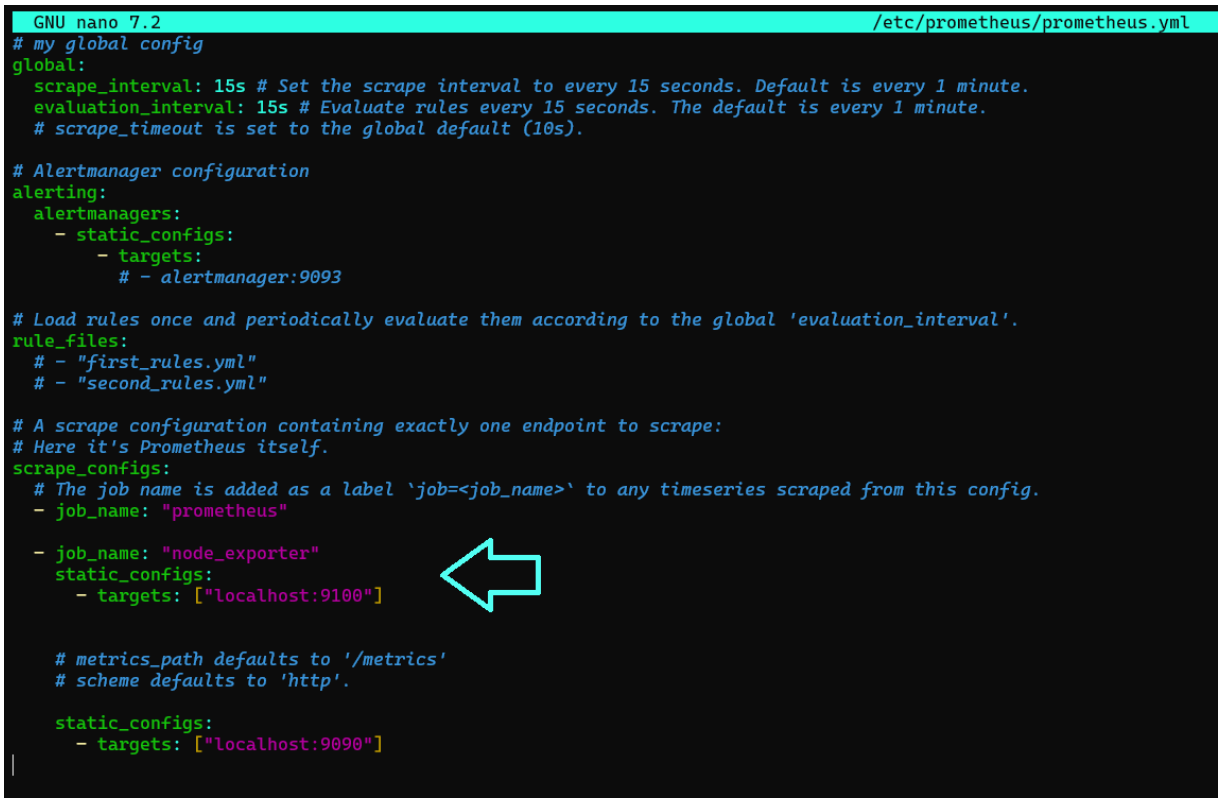

Figure 12: Edit prometheus.yml file to resolve Problem

And then I restarted Prometheus with

```
mahmoud@nextcloud:~$ sudo systemctl restart prometheus
```

After successfully integrating Prometheus and Node Exporter on my Ubuntu server, I imported the “Node Exporter Full” dashboard into Grafana. This dashboard provides real-time visual insights into key system metrics, including:

- CPU usage and load
- Memory and swap usage
- Disk usage and I/O
- Network traffic
- System uptime and

status See Figure 13.

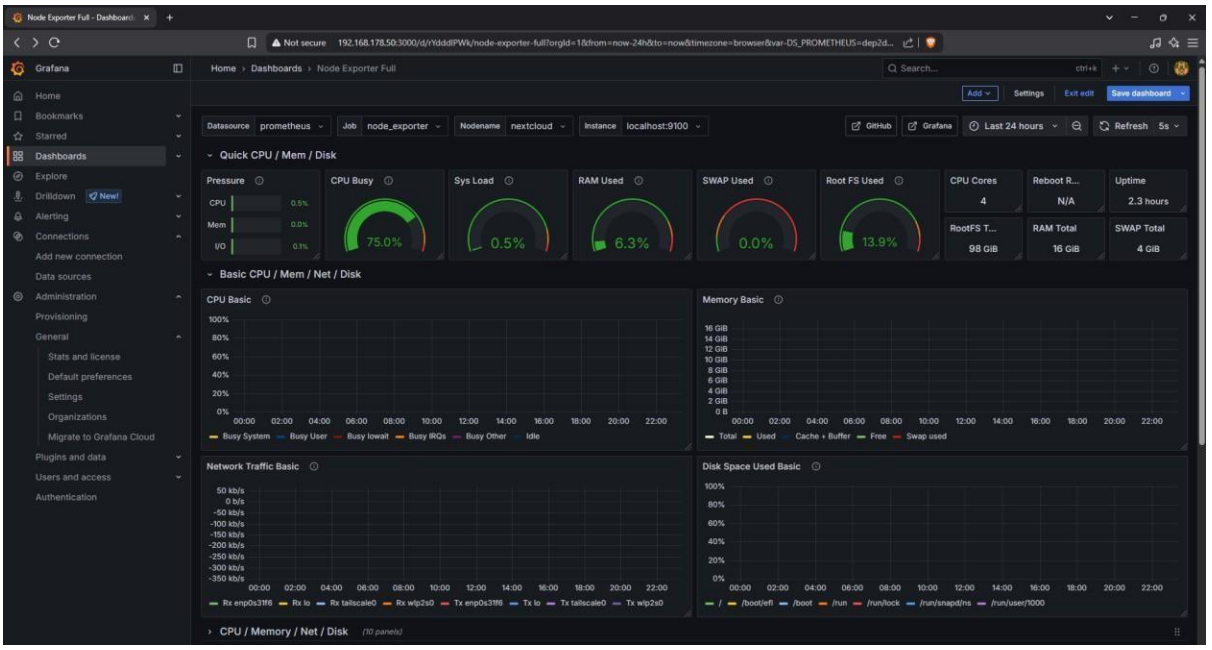

Figure 13: The customizable dashboard in Grafana