

# Project Description

Monitoring the CRUD application created and managed in part 01 of the project by Integrating Prometheus and Grafana for monitoring and virtualization

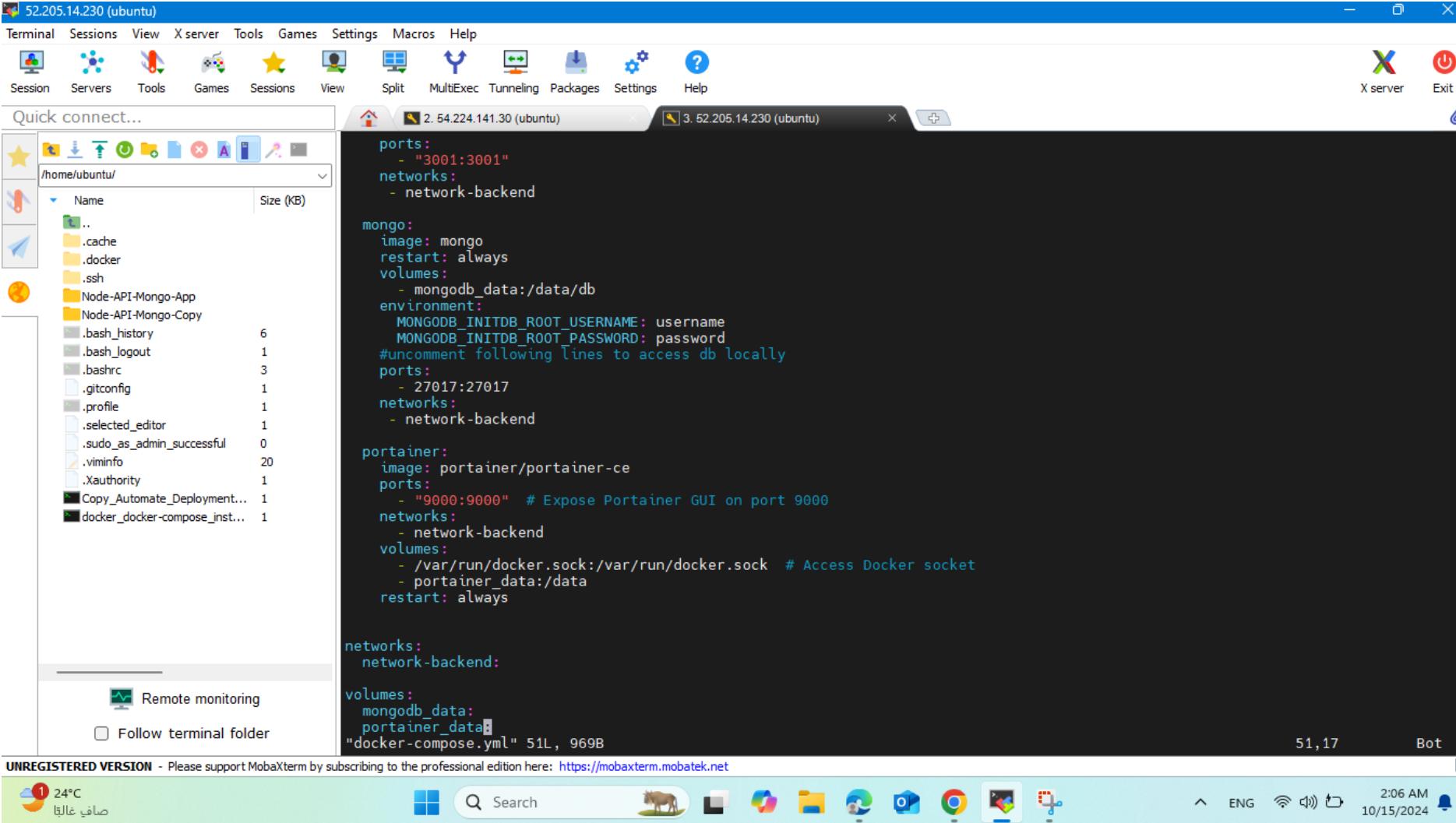
Part 02

# Monitoring

Step I

# **Integrating Prometheus for Monitoring**

# 1- Add portainer service in the docker-compose.yml file to run portainer



The screenshot shows a terminal window in MobaXterm with three tabs open:

- Tab 1: 52.205.14.230 (ubuntu) - Shows a file browser view of the user's home directory (`/home/ubuntu/`). The left sidebar includes icons for Session, Servers, Tools, Games, Sessions, View, Split, MultiExec, Tunneling, Packages, Settings, and Help. The right sidebar includes icons for X server and Exit.
- Tab 2: 2. 54.224.141.30 (ubuntu)
- Tab 3: 3. 52.205.14.230 (ubuntu)

The main terminal area displays the `docker-compose.yml` configuration file:

```
version: '3.8'

services:
  mongo:
    image: mongo
    restart: always
    volumes:
      - mongodb_data:/data/db
    environment:
      MONGODB_INITDB_ROOT_USERNAME: username
      MONGODB_INITDB_ROOT_PASSWORD: password
    ports:
      - 27017:27017
    networks:
      - network-backend
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock # Access Docker socket
      - portainer_data:/data
    restart: always

  portainer:
    image: portainer/portainer-ce
    ports:
      - "9000:9000" # Expose Portainer GUI on port 9000
    networks:
      - network-backend
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock # Access Docker socket
      - portainer_data:/data
    restart: always

networks:
  network-backend:

volumes:
  mongodb_data:
  portainer_data:
"docker-compose.yml" 51L, 969B
```

At the bottom of the terminal, there is a status bar with the following information:

- UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>
- 24°C صاف غالباً
- Search icon
- Cloud icon with 1
- File explorer icon
- Terminal icon
- Google Chrome icon
- File icon
- Folder icon
- Mail icon
- Image icon
- Document icon
- Help icon
- ENG
- Wi-Fi icon
- Speaker icon
- Bot icon
- 2:06 AM
- 10/15/2024

## 2-- Portainer Successfully running on port 9000

The screenshot shows a Microsoft Edge browser window with the title "Portainer". The address bar indicates the URL is [52.205.14.230:9000/#/init/admin](http://52.205.14.230:9000/#/init/admin). The main content is titled "New Portainer installation" and asks to create an initial administrator user. It has three input fields: "Username" (admin), "Password" (redacted), and "Confirm password" (redacted). Below these is a note: "⚠ The password must be at least 12 characters long." A blue "Create user" button is visible. At the bottom, there is a checked checkbox for "Allow collection of anonymous statistics. You can find more information about this in our [privacy policy](#)".

New Portainer installation

Please create the initial administrator user.

Username: admin

Password: .....

Confirm password: .....

⚠ The password must be at least 12 characters long. ✓

Create user

Allow collection of anonymous statistics. You can find more information about this in our [privacy policy](#).

> Restore Portainer from backup

24°C صافي غالباً

Search

2:05 AM 10/15/2024

### 3- Portainer Successfully running on port 9000

Portainer

Not secure 52.205.14.230:9000/#/wizard

Gmail Console Home | Co... Home - Microsoft A... (3479) DevOpsified |... DevOps Beginners t... (3004) All About RH... DevOps Track - Goo... Your Repositories All Bookmarks

Upgrade to Business Edition

portainer.io COMMUNITY EDITION

Home

Environment: None selected

Administration

User-related

Environment-related

Registries

Logs

Notifications

Settings

portainer.io Community Edition 2.21.3

Environment Wizard

Quick Setup

admin

Environment Wizard

Welcome to Portainer

We have connected your local environment of Docker to Portainer.

Get started below with your local portainer or connect more container environments.

Get Started

Add Environments

Proceed using the local environment which Portainer is running in

Connect to other environments

1 24°C صافي غالباً

Search

2:07 AM 10/15/2024

# 4- Docker in portainer

The screenshot shows the Portainer.io interface running in a web browser. The URL in the address bar is `52.205.14.230:9000/#!/environments`. The browser's toolbar includes icons for back, forward, search, and various extensions.

The Portainer sidebar on the left is titled "portainer.io COMMUNITY EDITION". It features a dark blue theme with white text. The "Environments" section is currently selected under "Environment-related". Other sections include "User-related", "Groups", "Tags", "Registries", "Logs", and "Notifications". The footer of the sidebar indicates "portainer.io Community Edition 2.21.3".

The main content area is titled "Environments" and displays a table of environments. The table has columns: Name, Type, URL, Group Name, and Actions. One environment is listed:

Name	Type	URL	Group Name	Actions
local	Docker	unix:///var/run/docker.sock	Unassigned	<a href="#">Manage access</a>

At the bottom right of the main content area, there is a "Items per page" dropdown set to 10.

The taskbar at the bottom of the screen shows several pinned icons: a weather icon (24°C), a search bar, a horse icon, a file folder, a user profile, a Google Chrome icon, a Microsoft Edge icon, a Microsoft Word icon, and a Microsoft Excel icon. The system tray on the right shows the date (10/15/2024), time (2:08 AM), battery level, signal strength, and a notification bell.

# 4- Docker in portainer

The screenshot shows the Portainer.io Community Edition dashboard for a local environment. The left sidebar lists navigation options: Home, Dashboard (selected), Templates, Stacks, Containers, Images, Networks, Volumes, Events, and Host. The main content area displays the following information:

**Environment info**

Environment	local (1 CPU, 1 GB - Standalone 27.3.1)
URL	/var/run/docker.sock
GPU	none
Tags	-

**Resource Statistics**

1 Stack	4 Containers
8 Images	15 Volumes

Details for containers: 4 running, 0 stopped, 0 healthy, 0 unhealthy.

At the bottom, the taskbar shows icons for Search, File Explorer, Task View, Google Chrome, and Edge, along with system status indicators for battery, signal, and volume.

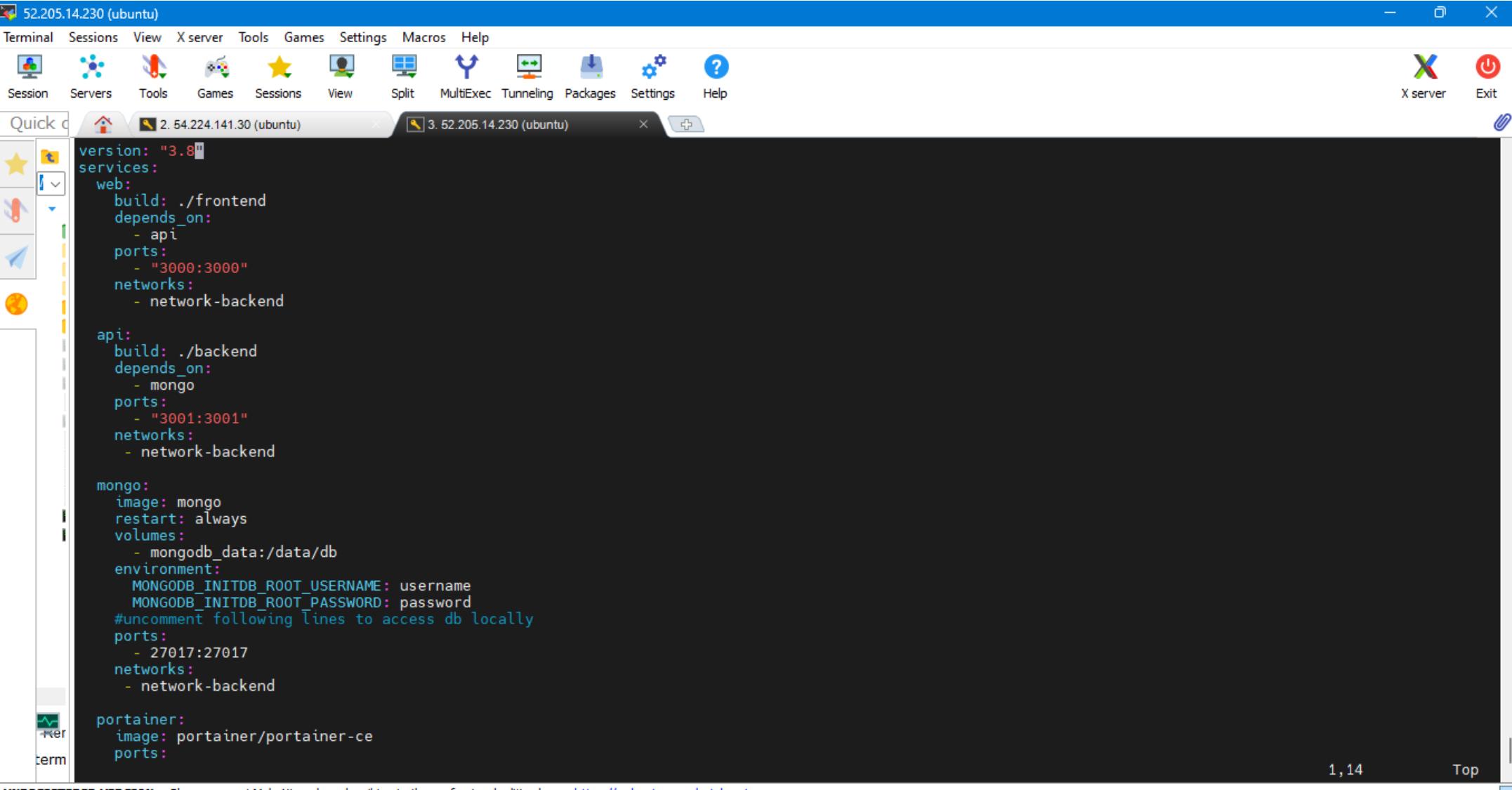
# 4- Docker containers in portainer

The screenshot shows the Portainer.io interface running in a browser window. The title bar indicates the URL is 52.205.14.230:9000/#!/2/docker/containers. The sidebar on the left is titled 'local' and includes links for Home, Dashboard, Templates, Stacks, Containers (which is selected), Images, Networks, Volumes, Events, and Host. The main content area is titled 'Container list' and displays a table of running containers. The table columns are: Name, State, Quick Actions, Stack, Image, Created, IP Address, and Published Port. The table contains five rows:

Name	State	Quick Actions	Stack	Image	Created	IP Address	Published Port
node-api-mongo-app_api_1	running	Restart, Stop, Kill, Restart, Pause, Resume, Remove	node-api-mongo-app	node-api-mongo-app_api	2024-10-15 02:01:09	172.18.0.4	3001:3001
node-api-mongo-app_mongo_1	running	Restart, Stop, Kill, Restart, Pause, Resume, Remove	node-api-mongo-app	mongo	2024-10-15 02:01:08	172.18.0.3	27017:27017
node-api-mongo-app_portainer_1	running	Restart, Stop, Kill, Restart, Pause, Resume, Remove	node-api-mongo-app	portainer/portainer-ce	2024-10-15 02:01:08	172.18.0.2	9000:9000
node-api-mongo-app_web_1	running	Restart, Stop, Kill, Restart, Pause, Resume, Remove	node-api-mongo-app	node-api-mongo-app_web	2024-10-15 02:01:10	172.18.0.5	3000:3000

At the bottom of the interface, there is a toolbar with icons for various functions like search, file operations, and monitoring. The status bar at the bottom right shows the date and time as 10/15/2024 2:11 AM.

# 5- Adding service in the docker-compose.yml file to run portainer



The screenshot shows a terminal window titled "52.205.14.230 (ubuntu)" in MobaXterm. The window displays a Docker Compose configuration file named `docker-compose.yml`. The file defines four services: `web`, `api`, `mongo`, and `portainer`. The `web` service uses a local build of the `frontend` application and depends on the `api` service. It maps port 3000 to 3000. The `api` service uses a local build of the `backend` application and depends on the `mongo` service. It maps port 3001 to 3001. The `mongo` service uses the official MongoDB image and restarts always, mapping port 27017 to 27017. The `portainer` service runs the official Portainer CE image and maps port 9443 to 9443. All services are connected to a shared network named `network-backend`.

```
version: "3.8"
services:
  web:
    build: ./frontend
    depends_on:
      - api
    ports:
      - "3000:3000"
    networks:
      - network-backend

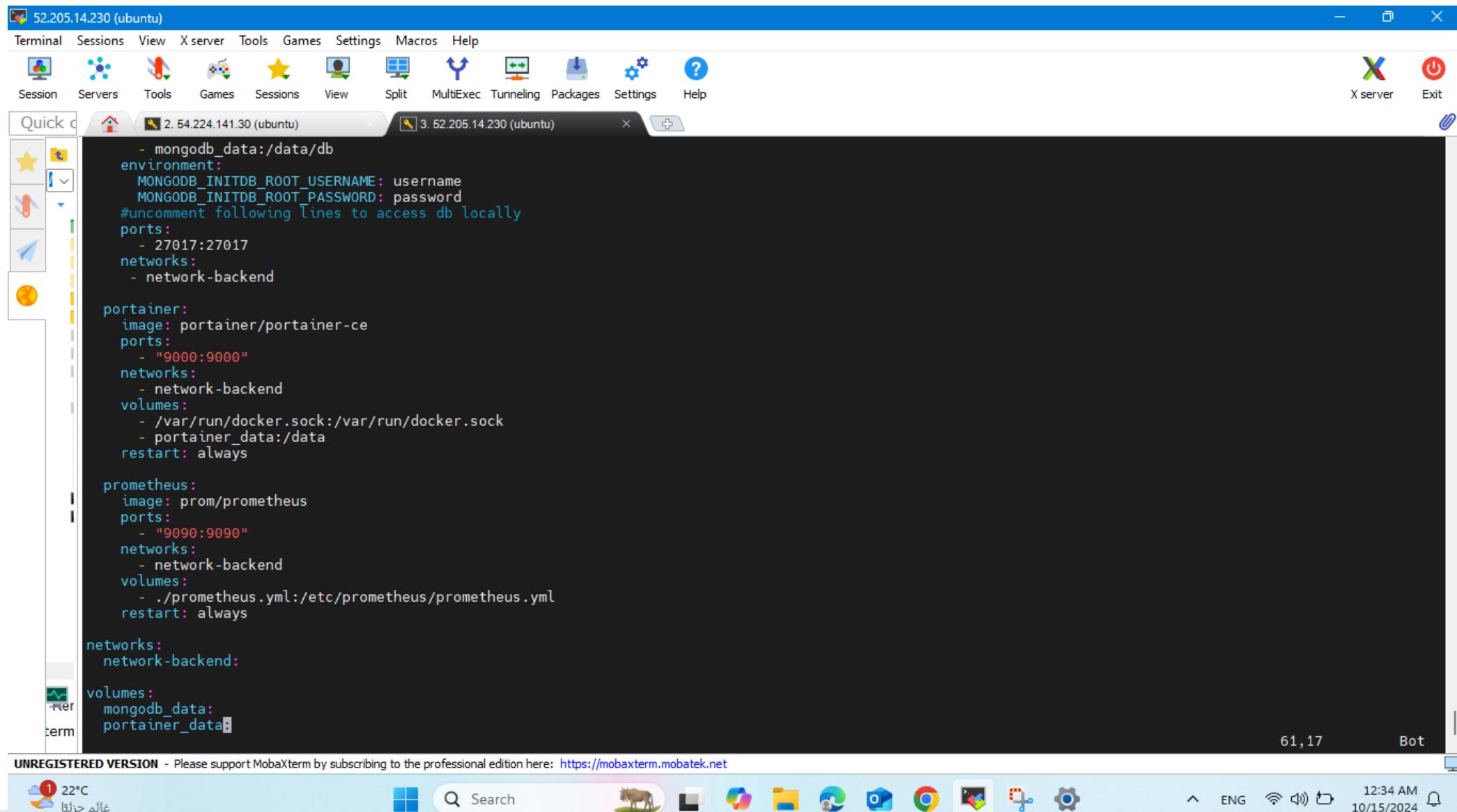
  api:
    build: ./backend
    depends_on:
      - mongo
    ports:
      - "3001:3001"
    networks:
      - network-backend

  mongo:
    image: mongo
    restart: always
    volumes:
      - mongodb_data:/data/db
    environment:
      MONGODB_INITDB_ROOT_USERNAME: username
      MONGODB_INITDB_ROOT_PASSWORD: password
    #uncomment following lines to access db locally
    ports:
      - 27017:27017
    networks:
      - network-backend

  portainer:
    image: portainer/portainer-ce
    ports:
```

At the bottom of the terminal window, there is a message: "UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>". The status bar at the bottom right shows system information including the date and time.

# 5- Adding service in the docker-compose.yml file to run portainer



The screenshot shows a MobaXterm session titled "52.205.14.230 (ubuntu)". The terminal window displays a Docker Compose configuration file (docker-compose.yml) with the following content:

```
version: '3.8'

services:
  mongodb:
    image: mongo:4.4
    environment:
      MONGODB_INITDB_ROOT_USERNAME: username
      MONGODB_INITDB_ROOT_PASSWORD: password
    #uncomment following lines to access db locally
    ports:
      - 27017:27017
    networks:
      - network-backend

  portainer:
    image: portainer/portainer-ce
    ports:
      - "9000:9000"
    networks:
      - network-backend
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock
      - portainer_data:/data
    restart: always

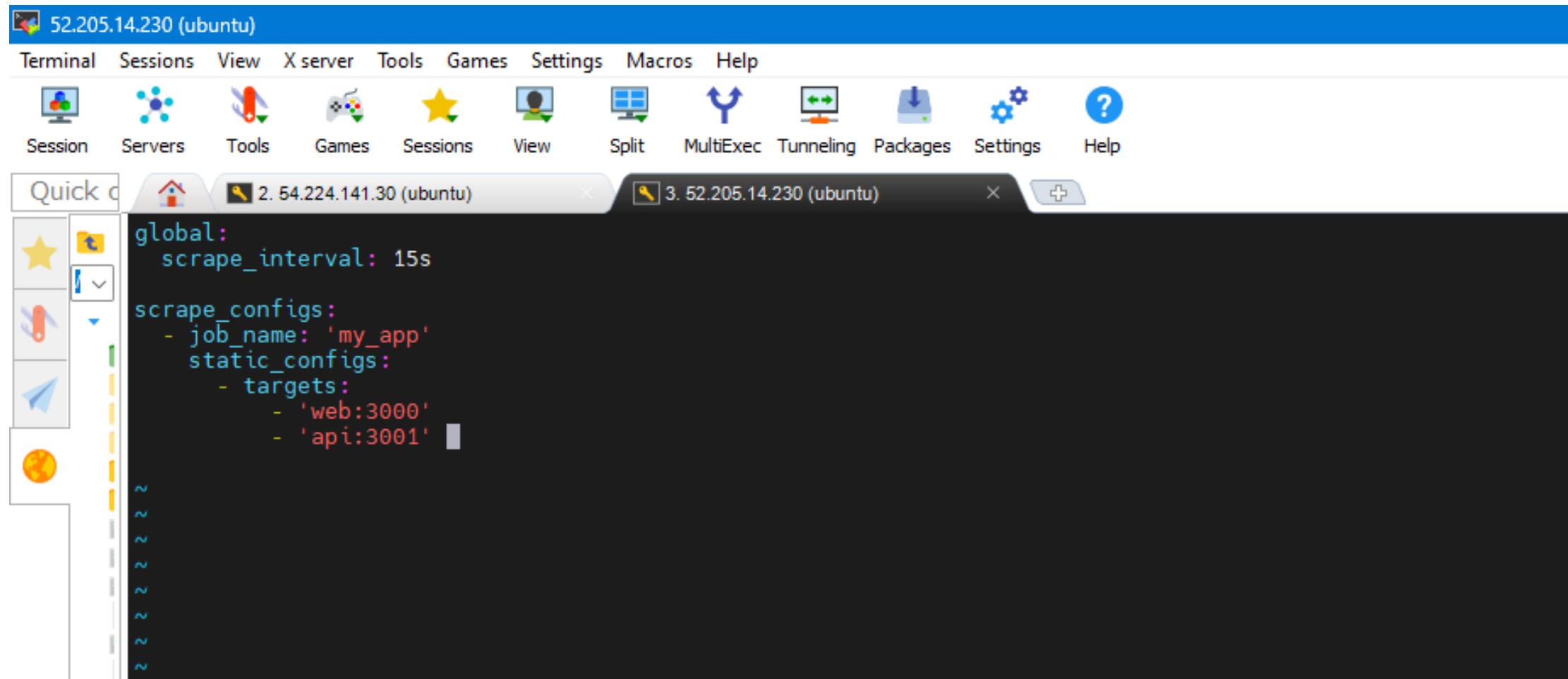
  prometheus:
    image: prom/prometheus
    ports:
      - "9090:9090"
    networks:
      - network-backend
    volumes:
      - ./prometheus.yml:/etc/prometheus/prometheus.yml
    restart: always

networks:
  network-backend:

volumes:
  mongodb_data:
  portainer_data:
```

The terminal window also shows the status bar with "61,17 Bot" and the bottom navigation bar with various icons.

## 6- Prometheus.yml file with targets declaration



The screenshot shows a desktop interface with a blue header bar. The title bar of the active window displays "52.205.14.230 (ubuntu)". The menu bar includes "Terminal", "Sessions", "View", "X server", "Tools", "Games", "Settings", "Macros", and "Help". Below the menu is a toolbar with icons for "Session", "Servers", "Tools", "Games", "Sessions", "View", "Split", "MultiExec", "Tunneling", "Packages", "Settings", and a question mark icon. The main window contains a terminal session with the following content:

```
global:
  scrape_interval: 15s

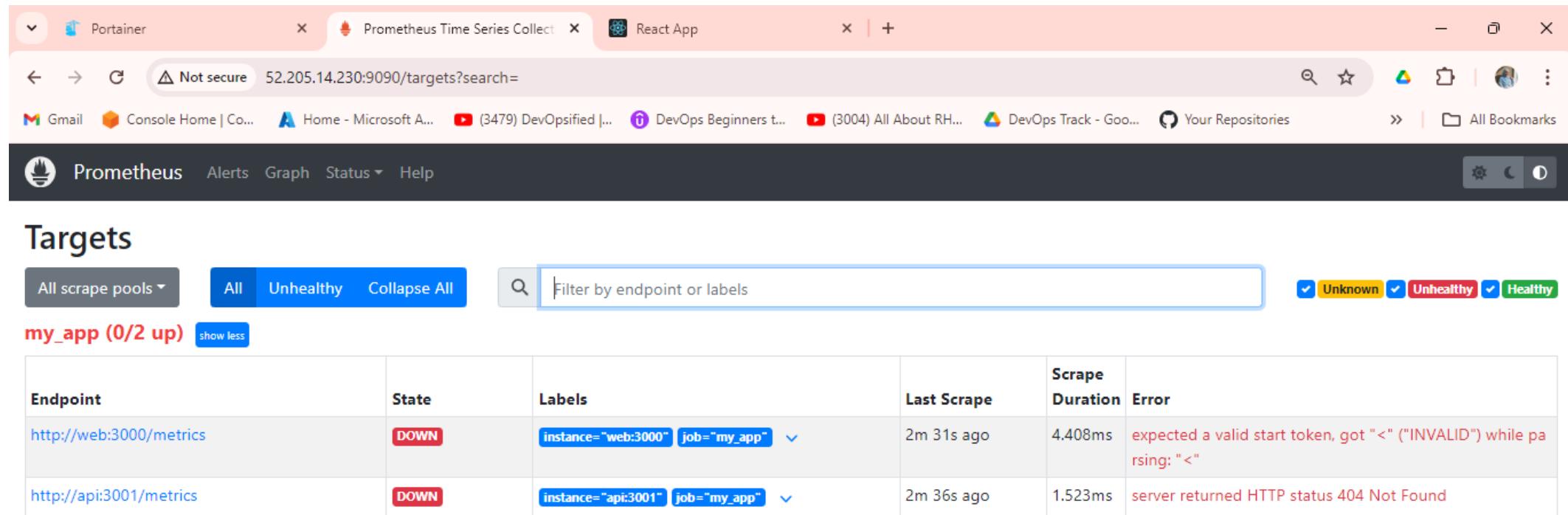
scrape_configs:
  - job_name: 'my_app'
    static_configs:
      - targets:
          - 'web:3000'
          - 'api:3001'
```

# 7- An error in time synchronization between the server and browser

The screenshot shows a web browser window with the following details:

- Browser Tabs:** Portainer, Prometheus Time Series Collector (active tab), React App.
- Address Bar:** Not secure 52.205.14.230:9090/graph?g0.expr=&g0.tab=1&g0.display\_mode=lines&g0.show\_exemplars=0&g0.range\_input=1h
- Toolbar:** Includes links to Gmail, Console Home, Microsoft Azure, YouTube, DevOps Beginners, DevOps Track, and Your Repositories, along with All Bookmarks.
- Prometheus UI Header:** Prometheus, Alerts, Graph, Status, Help.
- Configuration Options:** Use local time (unchecked), Enable query history (unchecked), Enable autocomplete (checked), Enable highlighting (checked), Enable linter (checked).
- Warning Message:** Warning: Error fetching server time: Detected 148.08500003814697 seconds time difference between your browser and the server. Prometheus relies on accurate time and time drift might cause unexpected query results.
- Search Bar:** Expression (press Shift+Enter for newlines) with Execute button.
- Panel Selection:** Graph selected over Table.
- Time Selection:** Evaluation time with back and forward arrows.
- Result Area:** No data queried yet.
- Buttons:** Remove Panel, Add Panel.

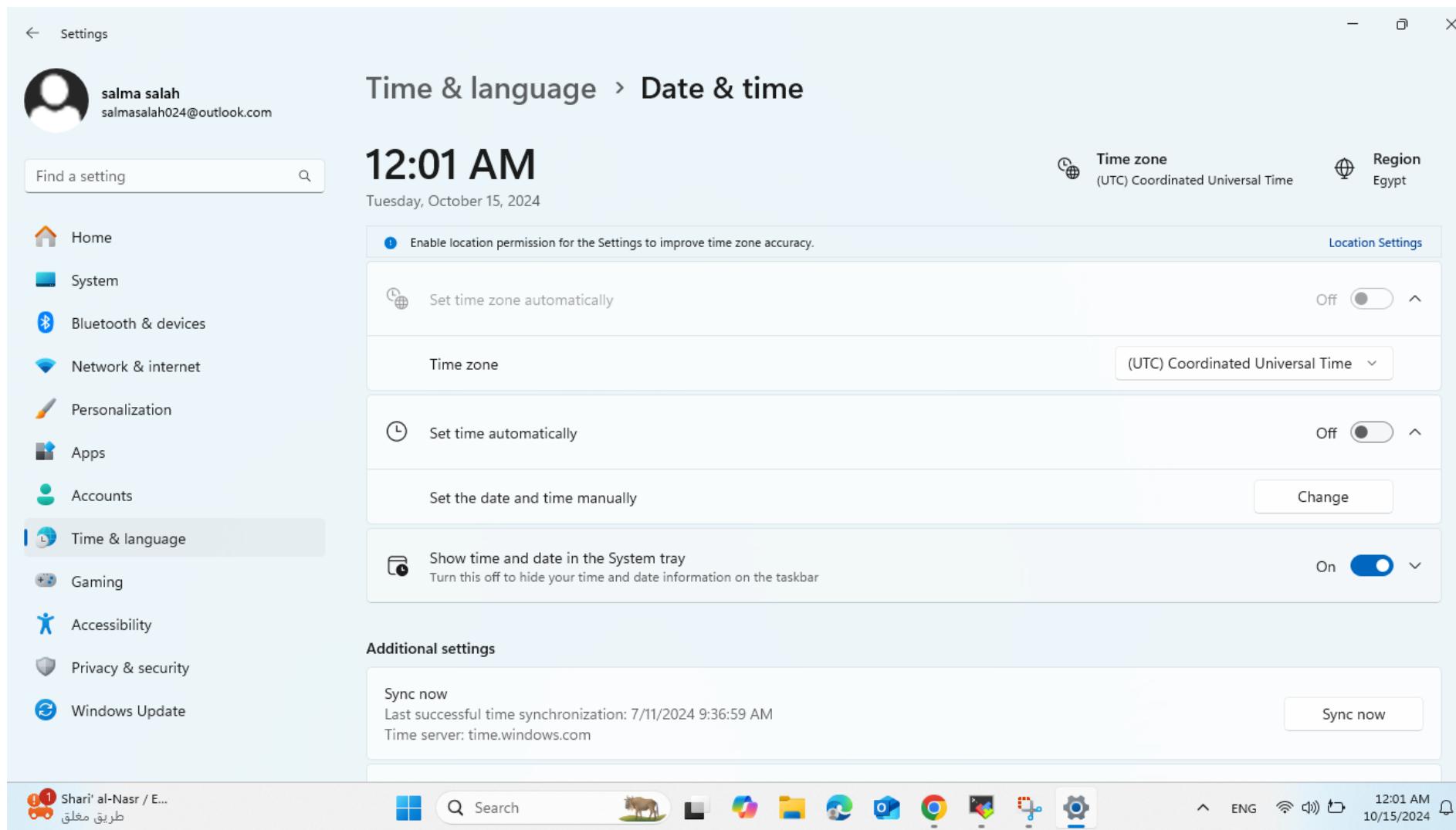
# 7- An error in time synchronization between the server and browser



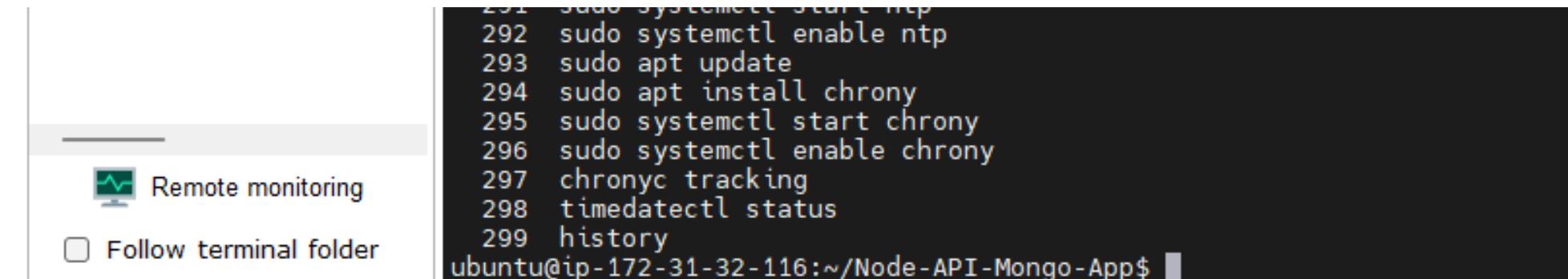
The screenshot shows a browser window with three tabs: Portainer, Prometheus Time Series Collect, and React App. The Prometheus tab is active, displaying the 'Targets' page. The page title is 'Targets' and it includes filters for 'All scrape pools' (dropdown), 'All' (selected), 'Unhealthy', and 'Collapse All'. A search bar allows filtering by endpoint or labels, with a dropdown menu showing 'instance="web:3000" job="my\_app"'. There are checkboxes for 'Unknown', 'Unhealthy' (checked), and 'Healthy'. The main table lists two targets:

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
<a href="#">http://web:3000/metrics</a>	DOWN	instance="web:3000" job="my_app"	2m 31s ago	4.408ms	expected a valid start token, got < ("INVALID") while parsing: <
<a href="#">http://api:3001/metrics</a>	DOWN	instance="api:3001" job="my_app"	2m 36s ago	1.523ms	server returned HTTP status 404 Not Found

# 8- Changing system's time to be UTC as the server



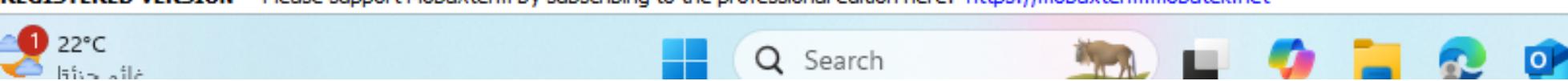
## 9- Installing chrony



The screenshot shows a terminal window in MobaXterm. On the left, there's a sidebar with a 'Remote monitoring' icon and a checkbox for 'Follow terminal folder'. The main terminal area displays the following command sequence:

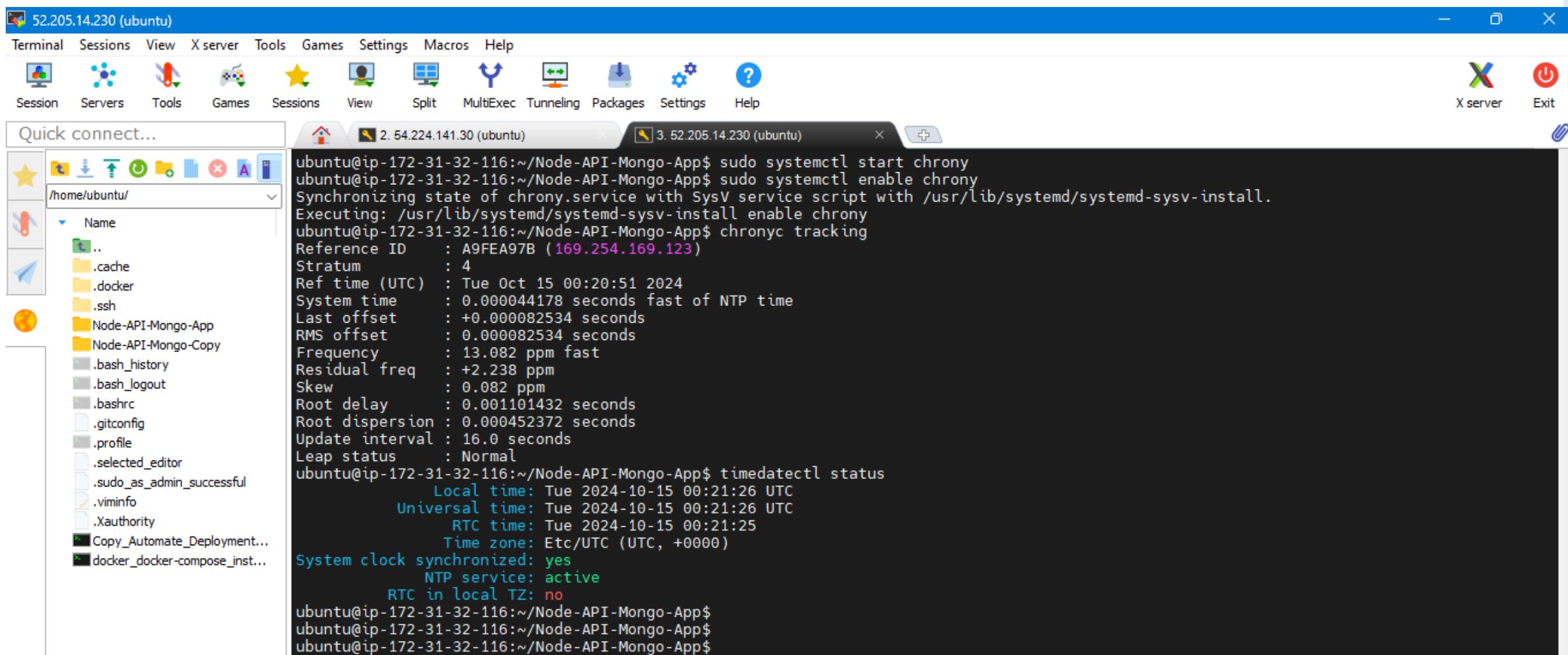
```
291 sudo systemctl start ntp  
292 sudo systemctl enable ntp  
293 sudo apt update  
294 sudo apt install chrony  
295 sudo systemctl start chrony  
296 sudo systemctl enable chrony  
297 chronyc tracking  
298 timedatectl status  
299 history  
ubuntu@ip-172-31-32-116:~/Node-API-Mongo-App$
```

At the bottom of the terminal window, there's a watermark: 'REGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>'.



The taskbar at the bottom of the screen shows several pinned icons: a weather widget (22°C), a file explorer, a search bar, a file manager, a messaging app, and a browser.

# 10- Checking synchronization



The screenshot shows a desktop environment with a terminal window open. The terminal window title is "3. 52.205.14.230 (ubuntu)". The terminal content displays the output of several commands related to system synchronization:

```
ubuntu@ip-172-31-32-116:~/Node-API-Mongo-App$ sudo systemctl start chrony
ubuntu@ip-172-31-32-116:~/Node-API-Mongo-App$ sudo systemctl enable chrony
Synchronizing state of chrony.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable chrony
ubuntu@ip-172-31-32-116:~/Node-API-Mongo-App$ chronyc tracking
Reference ID      : A9FEA97B (169.254.169.123)
Stratum          : 4
Ref time (UTC)   : Tue Oct 15 00:20:51 2024
System time      : 0.000044178 seconds fast of NTP time
Last offset      : +0.000082534 seconds
RMS offset       : 0.000082534 seconds
Frequency        : 13.082 ppm fast
Residual freq    : +2.238 ppm
Skew              : 0.082 ppm
Root delay       : 0.001101432 seconds
Root dispersion  : 0.000452372 seconds
Update interval  : 16.0 seconds
Leap status       : Normal
ubuntu@ip-172-31-32-116:~/Node-API-Mongo-App$ timedatectl status
          Local time: Tue 2024-10-15 00:21:26 UTC
          Universal time: Tue 2024-10-15 00:21:26 UTC
                    RTC time: Tue 2024-10-15 00:21:25
                   Time zone: Etc/UTC (UTC, +0000)
System clock synchronized: yes
                         NTP service: active
                RTC in local TZ: no
ubuntu@ip-172-31-32-116:~/Node-API-Mongo-App$
ubuntu@ip-172-31-32-116:~/Node-API-Mongo-App$
ubuntu@ip-172-31-32-116:~/Node-API-Mongo-App$
```

The terminal window is part of a desktop interface with a menu bar at the top and a sidebar on the left containing file and folder icons.

# 10- Checking synchronization

The screenshot shows a web browser window with multiple tabs open. The active tab is titled "What is my Timezone - WebBro" and displays the URL "webbrowsertools.com/timezone/". The page content is a table titled "Timezone info" with the following data:

Time on Server	Tue, 15 Oct 2024 01:21:20 GMT
Time on Local Machine	Tue Oct 15 2024 01:21:21 GMT+0000 (Coordinated Universal Time)
Time from Intl.DateTimeFormat #1	Tuesday, October 15, 2024 at 1:21:21 AM UTC
Time from Intl.DateTimeFormat #2	Tuesday, October 15, 2024 at 1:21:21 AM Coordinated Universal Time
Internet Beat	around @98
Timezone	UTC
Timezone Offset	0 minutes
Numbering System	latn
Locale	en-US
Calendar	gregory
Day	numeric
Month	numeric
Year	numeric
Milliseconds since January 1, 1970	1728955281153

At the bottom of the page, there is a cookie consent message: "We use cookies to enhance your experience. By continuing to visit this site you agree to our use of cookies. [Learn more](#)". There are "Got it!" and "Decline" buttons next to it.

The browser's taskbar at the bottom shows various pinned icons and the system tray with the date and time (10/15/2024, 1:21 AM).

# 11- Prometheus Successfully running

The screenshot shows a web browser window with the following details:

- Browser Tabs:** Portainer, React App, What is my Timezone - WebBro, Prometheus Time Series Collector.
- Address Bar:** Not secure 52.205.14.230:9090/graph?g0.expr=&g0.tab=1&g0.display\_mode=lines&g0.show\_exemplars=0&g0.range\_input=1h
- Toolbar:** Gmail, Console Home | Co..., Home - Microsoft A..., (3479) DevOpsified |..., DevOps Beginners t..., (3004) All About RH..., DevOps Track - Goo..., Your Repositories, All Bookmarks.
- Prometheus Interface:**
  - Header: Prometheus, Alerts, Graph, Status, Help.
  - Checkboxes: Use local time, Enable query history, Enable autocomplete (checked), Enable highlighting (checked), Enable linter (checked).
  - Search Bar: Expression (press Shift+Enter for newlines).
  - Buttons: Execute, Remove Panel.
  - Graph Tab: Evaluation time (with back and forward arrows).
  - Status: No data queried yet.
  - Buttons: Add Panel.
- Status Bar:** 22°C, 12:31 AM, ENG, 10/15/2024.

# 11- Prometheus Successfully running

The screenshot shows a browser window with the following tabs:

- Portainer | local
- React App
- What is my Timezone - WebBro
- Prometheus Time Series Collector

The main content area displays the Portainer.io interface for managing Docker containers. The sidebar on the left shows navigation links for Home, local, Dashboard, Templates, Stacks, Containers (which is selected), Images, Networks, Volumes, Events, and Host. The right panel shows a list of running containers:

Name	State	Image	Created	IP Address	Published Ports
node-api-mongo-app_api_1	running	node-api-mongo-app	2024-10-15 00:24:49	172.18.0.5	3001:3001
node-api-mongo-app_mongo_1	running	node-api-mongo-app	2024-10-15 00:24:48	172.18.0.3	27017:27017
node-api-mongo-app_portainer_1	running	node-api-mongo-app	2024-10-15 00:24:48	172.18.0.2	9000:9000
node-api-mongo-app_prometheus_1	running	node-api-mongo-app	2024-10-15 00:24:48	172.18.0.4	9090:9090
node-api-mongo-app_web_1	running	node-api-mongo-app	2024-10-15 00:24:50	172.18.0.6	3000:3000

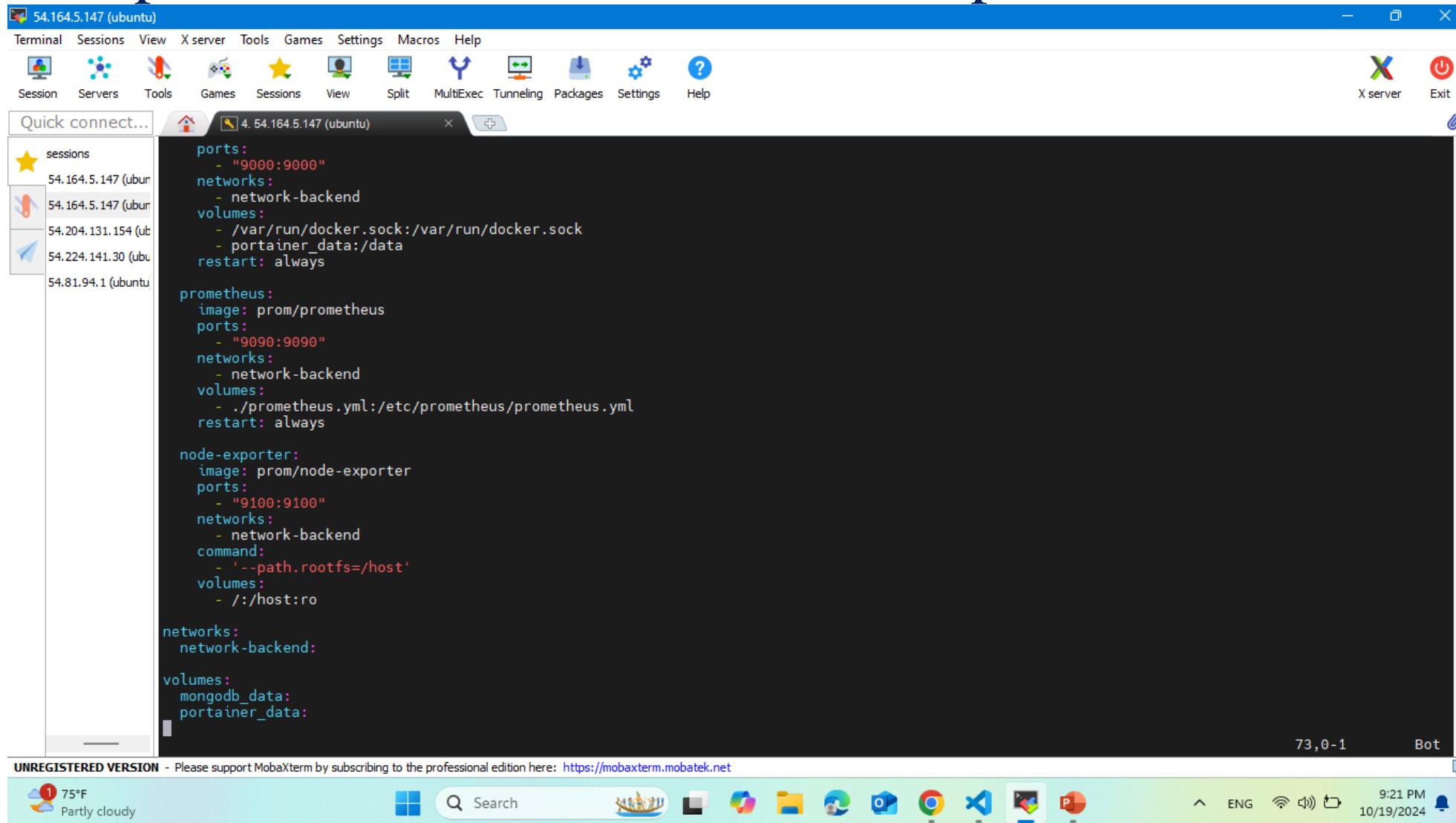
At the bottom, there is a system tray with icons for weather, battery, network, and system status, along with a search bar and a gear icon.

# Configuring Prometheus for Monitoring

- I. The host machine (AWS EC2) -> Node Exporter
- II. All the running containers -> cAdvisor
- III. The Application Metrics -> prom-client

## I- Using node exporter to collect host-ec2 metrics

# 12- Adding node-exporter to the (docker-compose.yml) file to be able to expose the host ec2 metrics for the prometheus



The screenshot shows a MobaXterm window titled "4. 54.164.5.147 (ubuntu)". The terminal content displays a Docker Compose configuration file:

```
54.164.5.147 (ubuntu)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
X server Exit
Quick connect...
sessions
54.164.5.147 (ubun
54.164.5.147 (ubun
54.204.131.154 (ub
54.224.141.30 (ub
54.81.94.1 (ubuntu

ports:
- "9000:9000"
networks:
- network-backend
volumes:
- /var/run/docker.sock:/var/run/docker.sock
- portainer_data:/data
restart: always

prometheus:
image: prom/prometheus
ports:
- "9090:9090"
networks:
- network-backend
volumes:
- ./prometheus.yml:/etc/prometheus/prometheus.yml
restart: always

node-exporter:
image: prom/node-exporter
ports:
- "9100:9100"
networks:
- network-backend
command:
- '--path.rootfs=/host'
volumes:
- /:/host:ro

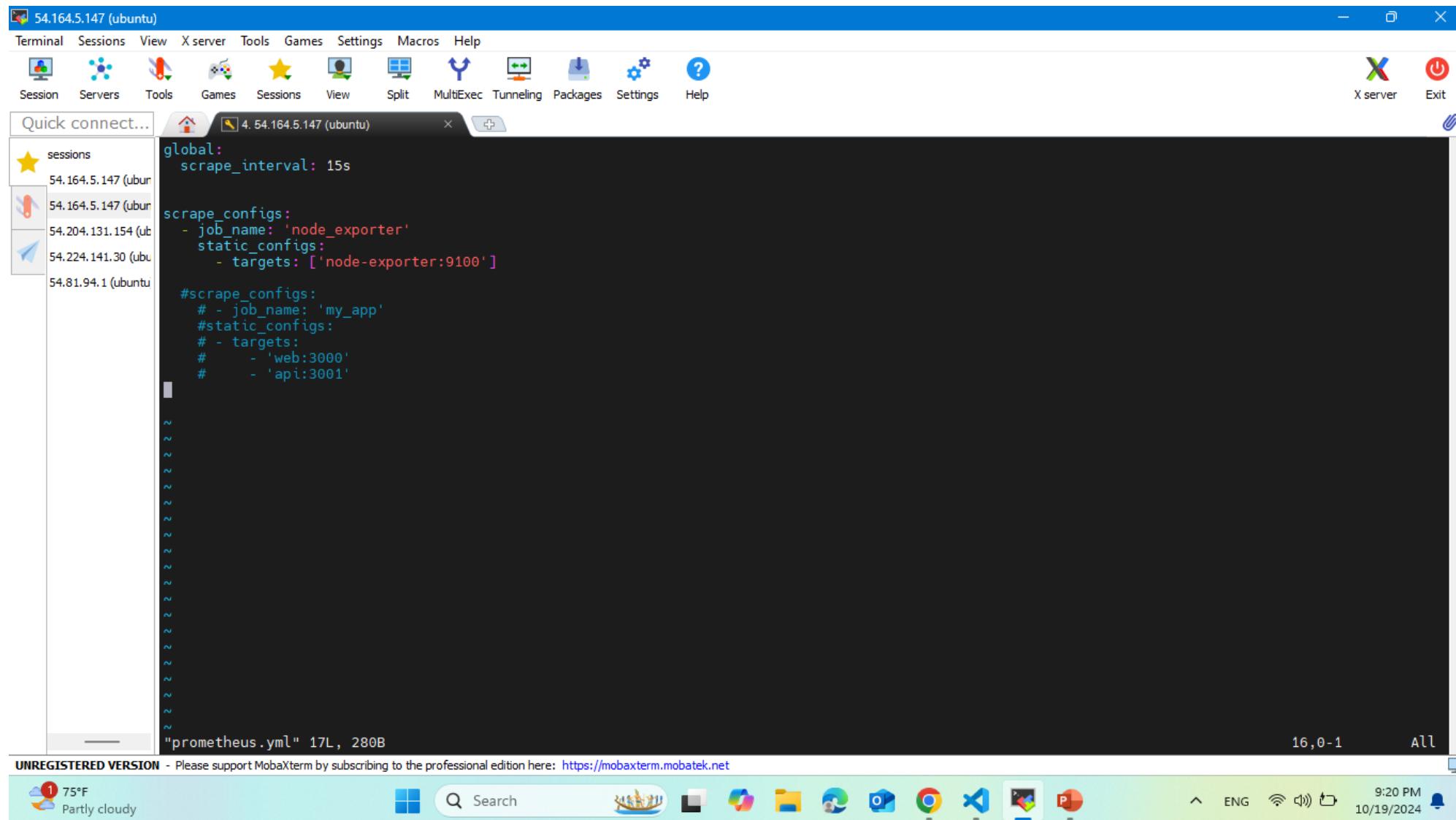
networks:
network-backend:

volumes:
mongodb_data:
portainer_data:
```

At the bottom of the terminal, there is a message: "UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>".

The system tray at the bottom right shows the date and time as "10/19/2024 9:21 PM".

# 13- Configuring node-exporter as a target in (Prometheus.yml) file



The screenshot shows a terminal window titled "54.164.5.147 (ubuntu)" running on MobaXterm. The window displays a portion of a Prometheus configuration file named "prometheus.yml". The configuration includes a global section with a scrape interval of 15 seconds, a scrape\_configs section with a job named "node\_exporter" using static configurations to target "node-exporter:9100", and a commented-out section for a job named "my\_app" with targets "web:3000" and "api:3001". The terminal interface includes a sidebar for session management and a status bar at the bottom.

```
global:
  scrape_interval: 15s

scrape_configs:
  - job_name: 'node_exporter'
    static_configs:
      - targets: ['node-exporter:9100']

#scrape_configs:
#  - job_name: 'my_app'
#    static_configs:
#      - targets:
#        - 'web:3000'
#        - 'api:3001'
```

"prometheus.yml" 17L, 280B

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

16,0-1 All

75°F Partly cloudy

Search

9:20 PM 10/19/2024

# 14- Node-exporter up in prometheus

The screenshot shows the Prometheus Targets page with the URL `54.164.5.147:9090/targets?search=node_exporter`. The page displays one target: `http://node-exporter:9100/metrics`, which is marked as **UP**. The **Labels** section shows discovered labels: `_address_= "node-exporter:9100"`, `_metrics_path_= "/metrics"`, `_scheme_= "http"`, `_scrape_interval_= "15s"`, `_scrape_timeout_= "10s"`, and `job= "node_exporter"`. The **Last Scrape** time is 2.304s ago, and the **Scrape Duration** is 11.317ms.

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
<a href="http://node-exporter:9100/metrics">http://node-exporter:9100/metrics</a>	UP	<code>instance= "node-exporter:9100"</code> <code>job= "node_exporter"</code> ^ <b>Discovered labels:</b> <code>_address_= "node-exporter:9100"</code> <code>_metrics_path_= "/metrics"</code> <code>_scheme_= "http"</code> <code>_scrape_interval_= "15s"</code> <code>_scrape_timeout_= "10s"</code> <code>job= "node_exporter"</code>	2.304s ago	11.317ms	

# 15- Running Queries

The screenshot shows the Prometheus Time Series Collector interface running in a browser window. The URL is `54.164.5.147:9090/graph?g0.expr=&g0.tab=1&g0.display_mode=lines&g0.show_exemplars=0&g0.range_input=1h`. The search bar contains the prefix "nd". A tooltip "ARP entries by device" points to the first result, "node\_arp\_entries". The results table lists various metrics under the "Table" section:

Table	Series	Type
No d	node_arp_entries	gauge
	node_boot_time_seconds	gauge
	node_context_switches_total	counter
	node_cooling_device_cur_state	gauge
	node_cooling_device_max_state	gauge
	node_cpu_guest_seconds_total	counter
	node_cpu_seconds_total	counter
	node_disk_discard_time_seconds_total	counter
	node_disk_discarded_sectors_total	counter
	node_disk_discards_completed_total	counter
	node_disk_discards_merged_total	counter
	node_disk_flush_requests_time_seconds_total	counter
	node_disk_flush_requests_total	counter
	node_disk_info	gauge
	node_disk_io_now	gauge
	node_disk_io_time_seconds_total	counter

At the bottom right of the interface, there is a "Remove Panel" button.



# 16- Monitoring ec2's CPU

The screenshot shows the Prometheus Time Series Collector interface. The URL in the browser is `54.164.5.147:9090/graph?g0.expr=node_cpu_seconds_total&g0.tab=1&g0.display_mode=lines&g0.show_exemplars=0&g0.range_input=...`. The interface includes a navigation bar with tabs for Prometheus, Alerts, Graph, Status, and Help. Below the navigation bar are several checkboxes: Use local time, Enable query history, Enable autocomplete (checked), Enable highlighting (checked), and Enable linter. A search bar contains the query `node_cpu_seconds_total`. To the right of the search bar are buttons for Table (selected) and Graph, Evaluation time, and an Execute button. The results table shows the following data:

Series	Value
<code>node_cpu_seconds_total(cpu="0", instance="node-exporter:9100", job="node_exporter", mode="idle")</code>	6265.4
<code>node_cpu_seconds_total(cpu="0", instance="node-exporter:9100", job="node_exporter", mode="iowait")</code>	43.57
<code>node_cpu_seconds_total(cpu="0", instance="node-exporter:9100", job="node_exporter", mode="irq")</code>	0
<code>node_cpu_seconds_total(cpu="0", instance="node-exporter:9100", job="node_exporter", mode="nice")</code>	0.67
<code>node_cpu_seconds_total(cpu="0", instance="node-exporter:9100", job="node_exporter", mode="softirq")</code>	0.62
<code>node_cpu_seconds_total(cpu="0", instance="node-exporter:9100", job="node_exporter", mode="steal")</code>	249.02
<code>node_cpu_seconds_total(cpu="0", instance="node-exporter:9100", job="node_exporter", mode="system")</code>	29.87
<code>node_cpu_seconds_total(cpu="0", instance="node-exporter:9100", job="node_exporter", mode="user")</code>	93.76

Below the table are buttons for Remove Panel and Add Panel.



# 17- All containers successfully running on portainer

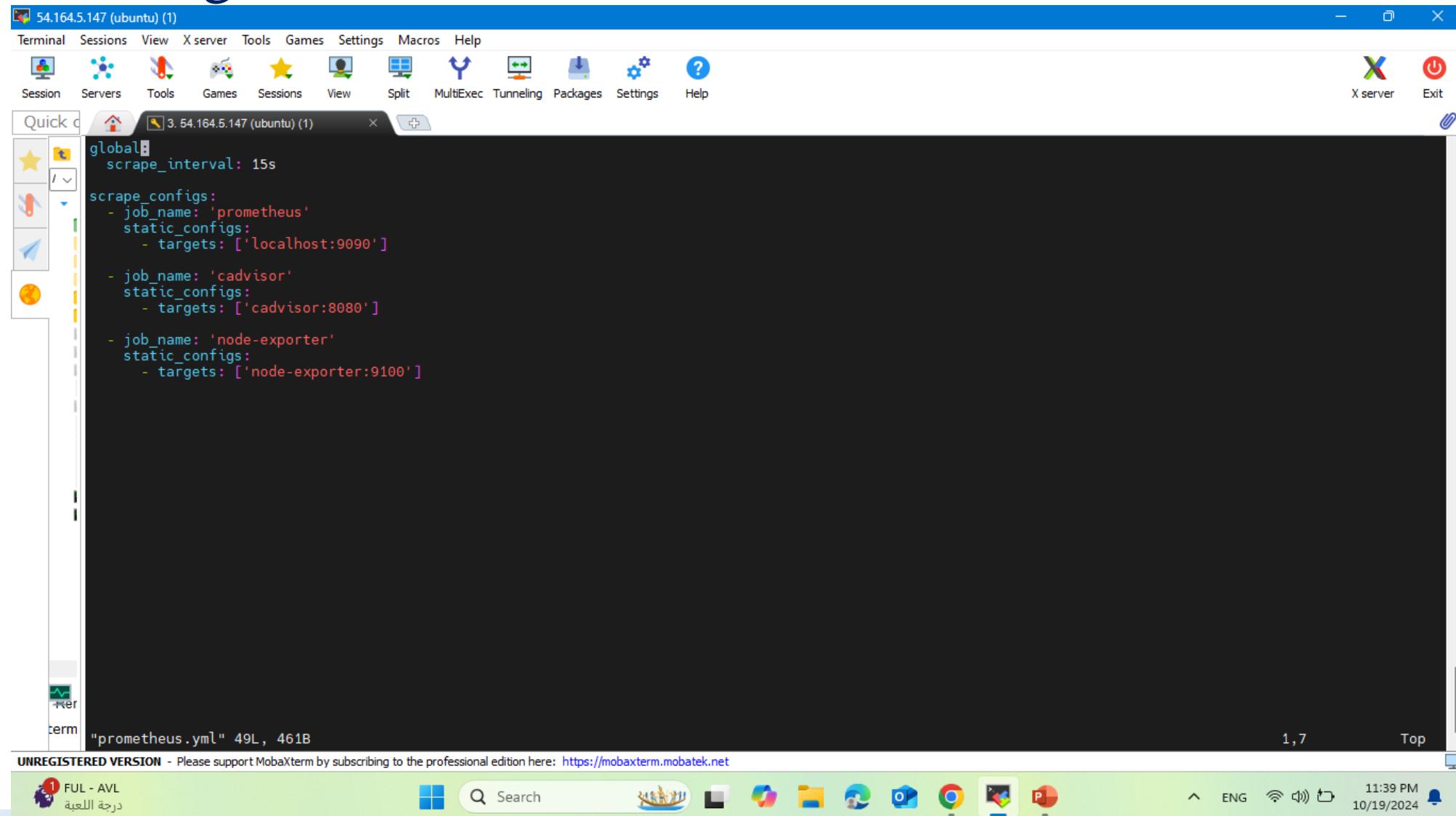
The screenshot shows the Portainer interface for managing Docker containers. The left sidebar is titled 'portainer.io COMMUNITY EDITION' and includes sections for Home, local (Dashboard, Templates, Stacks, Containers, Images, Networks, Volumes, Events, Host), Administration, and a footer note about version 2.21.3. The main content area is titled 'Container list' and displays a table of running containers. The table columns are: Name, State, Quick Actions, Stack, Image, Created, IP Address, and Published Ports. There are 7 containers listed:

Name	State	Quick Actions	Stack	Image	Created	IP Address	Published Ports
node-api-mongo-app_api_1	running	Stop, Start, Kill, Restart, Pause, Resume, Remove	node-api-mongo-app	node-api-mongo-app_api	2024-10-19 21:15:23	172.18.0.6	3001:3001
node-api-mongo-app_mongo_1	running	Stop, Start, Kill, Restart, Pause, Resume, Remove	node-api-mongo-app	mongo	2024-10-19 21:15:21	172.18.0.3	27017:27017
node-api-mongo-app_node-exporter_1	running	Stop, Start, Kill, Restart, Pause, Resume, Remove	node-api-mongo-app	prom/node-exporter	2024-10-19 21:15:21	172.18.0.4	9100:9100
node-api-mongo-app_portainer_1	running	Stop, Start, Kill, Restart, Pause, Resume, Remove	node-api-mongo-app	portainer/portainer-ce	2024-10-19 21:15:21	172.18.0.2	9000:9000
node-api-mongo-app_prometheus_1	running	Stop, Start, Kill, Restart, Pause, Resume, Remove	node-api-mongo-app	prom/prometheus	2024-10-19 21:15:21	172.18.0.5	9090:9090
node-api-mongo-app_web_1	running	Stop, Start, Kill, Restart, Pause, Resume, Remove	node-api-mongo-app	node-api-mongo-app_web	2024-10-19 21:15:24	172.18.0.7	3000:3000

The browser address bar shows the URL as '54.164.5.147:9000#!/2/docker/containers'. The system tray at the bottom indicates a weather of 24°C and a date/time of 10/19/2024 at 10:12 PM.

## II- Using cAdvisor to collect containers' metrics

# 18- Adding a job in (Prometheus.yml) file and configuring cAdvisor as target



The screenshot shows a terminal window titled "54.164.5.147 (ubuntu) (1)" in the MobaXterm interface. The window contains the following Prometheus configuration file:

```
global:
  scrape_interval: 15s

scrape_configs:
  - job_name: 'prometheus'
    static_configs:
      - targets: ['localhost:9090']

  - job_name: 'cadvisor'
    static_configs:
      - targets: ['cadvisor:8080']

  - job_name: 'node-exporter'
    static_configs:
      - targets: ['node-exporter:9100']
```

The terminal window has a dark background and a light-colored sidebar on the left. The status bar at the bottom shows the file name "prometheus.yml" and its size "49L, 461B". The system tray at the bottom right includes icons for battery level (1,7), language (ENG), signal strength, and date/time (11:39 PM, 10/19/2024).

## 19- Adding cAdvisor as a service in Docker Compose File

The screenshot shows a desktop environment with a terminal window open. The terminal window title is "3. 54.164.5.147 (ubuntu) (1)". The content of the terminal window is a Docker Compose configuration file named docker-compose.yml:

```
version: '3.8'
services:
  - "9090:9090"
    networks:
    - network-backend
    volumes:
    - ./prometheus.yml:/etc/prometheus/prometheus.yml
    restart: always

  node-exporter:
    image: prom/node-exporter
    ports:
    - "9100:9100"
    networks:
    - network-backend
    command:
    - '--path.rootfs=/host'
    volumes:
    - /:/host:ro

  cAdvisor:
    image: gcr.io/cadvisor/cadvisor:v0.47.1
    ports:
    - "8080:8080"
    volumes:
    - ::/rootfs:ro
    - /var/run:/var/run:ro
    - /sys:/sys:ro
    - /var/lib/docker/:/var/lib/docker:ro
    networks:
    - network-backend

  networks:
    network-backend:

  volumes:
    mongodb_data:
    portainer_data:
```

**UNREGISTERED VERSION** - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net/>



# 20- cAdvisor Successfully added to targets

The screenshot shows a web browser window with the URL `54.164.5.147:9090/targets?search=`. The browser tabs include "React App", "Portainer", "Prometheus Time Series Collector", and a "+" tab. The bookmarks bar contains links to "Gmail", "Console Home | Co...", "Home - Microsoft A...", "(3479) DevOpsified |...", "(3479) DevOpsified |...", "(3004) All About RH...", "DevOps Track - Goo...", "Your Repositories", and "All Bookmarks". The main content area is titled "Targets" and displays three healthy targets: "cadvisor (1/1 up)", "node-exporter (1/1 up)", and "prometheus (1/1 up)". Each target entry includes a "show more" link. There are filters for "All scrape pools" (dropdown), "All", "Unhealthy", "Collapse All", and a search bar "Filter by endpoint or labels". Status filters include "Unknown", "Unhealthy", and "Healthy".



# 20- cAdvisor Successfully added to targets

The screenshot shows a web browser window with three tabs open:

- React App
- Portainer
- Prometheus Time Series Collector

The current view is the "Targets" page under the Prometheus tab. The page displays three groups of targets:

### cadvisor (1/1 up)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://cAdvisor:8080/metrics	UP	instance="cAdvisor:8080" job="cAdvisor"	42.52s ago	186.694ms	

### node-exporter (1/1 up)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://node-exporter:9100/metrics	UP	instance="node-exporter:9100" job="node-exporter"	44.927s ago	15.059ms	

### prometheus (1/1 up)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:9090/metrics	UP	instance="localhost:9090" job="prometheus"	42.270s ago	17.487ms	

At the bottom of the screen, there is a taskbar with various icons and system status information.

# 21- Querying on the containers' metrics

The screenshot shows the Prometheus Time Series Collector interface in a browser window. The URL is `54.164.5.147:9090/graph?g0.expr=&g0.tab=1&g0.display_mode=lines&g0.show_exemplars=0&g0.range_input=1h`. The search bar contains the prefix 'c'. A tooltip for the metric `cadvisor_version_info` is displayed, stating: "A metric with a constant '1' value labeled by kernel version, OS version, docker version, cAdvisor version & cAdvisor revision." The interface includes a sidebar with navigation links like Prometheus, Alerts, Graph, Status, and Help, and a bottom panel with various configuration and status indicators.

Table	Name	Type
No d	<code>cadvisor_version_info</code>	gauge
	<code>ceil</code>	function
	<code>changes</code>	function
	<code>clamp</code>	function
	<code>clamp_max</code>	function
	<code>clamp_min</code>	function
	<code>container_blkio_device_usage_total</code>	counter
	<code>container_cpu_load_average_10s</code>	gauge
	<code>container_cpu_system_seconds_total</code>	counter
	<code>container_cpu_usage_seconds_total</code>	counter
	<code>container_cpu_user_seconds_total</code>	counter
	<code>container_fs_inodes_free</code>	gauge
	<code>container_fs_inodes_total</code>	gauge
	<code>container_fs_io_current</code>	gauge
	<code>container_fs_io_time_seconds_total</code>	counter
	<code>container_fs_io_time_weighted_seconds_total</code>	counter

# 21- Querying on the containers' metrics

The screenshot shows a web browser window with three tabs open:

- React App
- Portainer
- Prometheus Time Series Collect (active tab)

The URL in the address bar is `54.164.5.147:9090/graph?g0.expr=container_cpu_load_average_10s&g0.tab=1&g0.display_mode=lines&g0.show_exemplars=0&g0.range_input=`. The browser interface includes a search bar, a toolbar with various icons, and a navigation bar with links to Gmail, Microsoft Console Home, and other DevOps resources.

The main content area is a Prometheus query interface with the following settings:

- Use local time:
- Enable query history:
- Enable autocomplete:
- Enable highlighting:
- Enable linter:

The query input field contains `container_cpu_load_average_10s`. The "Graph" tab is selected. The results table shows the following data:

Series	Value
<code>container_cpu_load_average_10s{id="/", instance="cadvisor:8080", job="cadvisor"}</code>	0
<code>container_cpu_load_average_10s{id="/init.scope", instance="cadvisor:8080", job="cadvisor"}</code>	0
<code>container_cpu_load_average_10s{id="/system.slice", instance="cadvisor:8080", job="cadvisor"}</code>	0
<code>container_cpu_load_average_10s{id="/system.slice/ModemManager.service", instance="cadvisor:8080", job="cadvisor"}</code>	0
<code>container_cpu_load_average_10s{id="/system.slice/acpid.service", instance="cadvisor:8080", job="cadvisor"}</code>	0
<code>container_cpu_load_average_10s{id="/system.slice/chrony.service", instance="cadvisor:8080", job="cadvisor"}</code>	0
<code>container_cpu_load_average_10s{id="/system.slice/cloud-init-local.service", instance="cadvisor:8080", job="cadvisor"}</code>	0
<code>container_cpu_load_average_10s{id="/system.slice/containerd.service", instance="cadvisor:8080", job="cadvisor"}</code>	0
<code>container_cpu_load_average_10s{id="/system.slice/cron.service", instance="cadvisor:8080", job="cadvisor"}</code>	0
<code>container_cpu_load_average_10s{id="/system.slice/dbus.service", instance="cadvisor:8080", job="cadvisor"}</code>	0
<code>container_cpu_load_average_10s{id="/system.slice/docker.service", instance="cadvisor:8080", job="cadvisor"}</code>	0
<code>container_cpu_load_average_10s{id="/system.slice/docker.socket", instance="cadvisor:8080", job="cadvisor"}</code>	0
<code>container_cpu_load_average_10s{id="/system.slice/lxd-installer.socket", instance="cadvisor:8080", job="cadvisor"}</code>	0
<code>container_cpu_load_average_10s{id="/system.slice/multipathd.service", instance="cadvisor:8080", job="cadvisor"}</code>	0

The status bar at the bottom shows system information: 23°C, ENG, 10:58 PM, 10/19/2024.

## 22- Filter a specific container to query

The screenshot shows the Prometheus Time Series Collector interface. In the top navigation bar, there are three tabs: "React App", "Portainer", and "Prometheus Time Series Collector". The current tab is "Prometheus Time Series Collector". The URL in the address bar is `54.164.5.147:9090/graph?g0.expr=container_fs_io_current&g0.tab=1&g0.display_mode=lines&g0.show_exemplars=0&g0.range_input=1h`. Below the address bar, there is a toolbar with icons for Gmail, Console Home, Microsoft Edge, YouTube, DevOpsified, DevOps Beginners, DevOps Track, and Your Repositories, along with a "All Bookmarks" button.

The main interface has a dark header with the Prometheus logo and navigation links for Alerts, Graph, Status, and Help. Below the header, there are several configuration checkboxes: "Use local time" (unchecked), "Enable query history" (unchecked), "Enable autocomplete" (checked), "Enable highlighting" (checked), and "Enable linter" (checked). A search bar contains the query `container_fs_io_current{image=""}`. To the right of the search bar are buttons for "Execute" and "Table" (which is selected) or "Graph". A dropdown menu is open over the search bar, listing various container images: "gcr.io/cadvisor/cadvisor:v0.47.1", "mongo", "node-api-mongo-app\_api", "node-api-mongo-app\_web", "portainer/portainer-ce", "prom/node-exporter", and "prom/prometheus". The "mongo" entry is highlighted. Below the dropdown, the query results are listed as a series of text lines, each representing a metric instance. The results are sorted by evaluation time, with the most recent at the top. The results show metrics for various devices and jobs, such as "/dev", "/dev/root", "/dev/shm", "/proc/acpi", "/proc/kcore", "/proc/keys", "/proc/latency\_stats", "/proc/scsi", "/proc/timer\_list", "/rootfs/dev/shm", "/rootfs/run", and "/rootfs/run/lock", all associated with the "cadvisor" job and the "cAdvisor" instance.

Load time: 172ms Resolution: 14s Result series: 44

(container\_fs\_io\_current{device="/dev", id="/", instance="cadvisor:8080", job="cadvisor"})  
(container\_fs\_io\_current{device="/dev/root", id="/", instance="cadvisor:8080", job="cadvisor"})  
(container\_fs\_io\_current{device="/dev/shm", id="/", instance="cadvisor:8080", job="cadvisor"})  
(container\_fs\_io\_current{device="/dev/xvda16", id="/", instance="cadvisor:8080", job="cadvisor"})  
(container\_fs\_io\_current{device="/proc/acpi", id="/", instance="cadvisor:8080", job="cadvisor"})  
(container\_fs\_io\_current{device="/proc/kcore", id="/", instance="cadvisor:8080", job="cadvisor"})  
(container\_fs\_io\_current{device="/proc/keys", id="/", instance="cadvisor:8080", job="cadvisor"})  
(container\_fs\_io\_current{device="/proc/latency\_stats", id="/", instance="cadvisor:8080", job="cadvisor"})  
(container\_fs\_io\_current{device="/proc/scsi", id="/", instance="cadvisor:8080", job="cadvisor"})  
(container\_fs\_io\_current{device="/proc/timer\_list", id="/", instance="cadvisor:8080", job="cadvisor"})  
(container\_fs\_io\_current{device="/rootfs/dev/shm", id="/", instance="cadvisor:8080", job="cadvisor"})  
(container\_fs\_io\_current{device="/rootfs/run", id="/", instance="cadvisor:8080", job="cadvisor"})  
(container\_fs\_io\_current{device="/rootfs/run/lock", id="/", instance="cadvisor:8080", job="cadvisor"})

23°C غائم جزئيا

Search

11:00 PM 10/19/2024

## 22- Filter a specific container to query

The screenshot shows a browser window with three tabs: "React App", "Portainer", and "Prometheus Time Series Collector". The "Prometheus Time Series Collector" tab is active, displaying a search bar with the query `container_fs_io_current{image="mongo"}`. Below the search bar, there are several configuration checkboxes: "Use local time" (unchecked), "Enable query history" (unchecked), "Enable autocomplete" (checked), "Enable highlighting" (checked), and "Enable linter" (checked). The results panel shows a single result set with the following details:

```
container_fs_io_current{container_label_com_docker_compose_config_hash="66e08cd78cf25c6e8fa3a11ef9b6f4d0873c93690f7d4b76452547787d6129ed", container_label_com_docker_compose_container_number="1", container_label_com_docker_compose_oneoff="False", container_label_com_docker_compose_project="node-api-mongo-app", container_label_com_docker_compose_project_config_files="docker-compose.yml", container_label_com_docker_compose_project_working_dir="/home/ubuntu/Node-API-Mongo-App", container_label_com_docker_compose_service="mongo", container_label_com_docker_compose_version="1.28.5", container_label_org_opencontainers_image_ref_name="ubuntu", container_label_org_opencontainers_image_version="24.04", device="/dev/root", id="/system.slice/docker-2c615e70ee8da8814ec736c05cd344cb86a0a2a89401afe1a039486a7b294fd8.scope", image="mongo", instance="cadvisor:8080", job="cadvisor", name="node-api-mongo-app_mongo_1"}
```

The results panel includes a "Remove Panel" button and a "Graph" tab. At the bottom left, there is an "Add Panel" button.



# 23- All containers successfully running on portainer

The screenshot shows the Portainer interface for managing Docker stacks. The left sidebar is titled 'local' and includes options for Dashboard, Templates, Stacks (which is selected), Containers, Images, Networks, Volumes, Events, and Host. The main panel displays 'Stack details' for 'node-api-mongo-app'. The 'Containers' section lists seven containers:

Name	Status	Image	Created	IP Address	Public IP
node-api-mongo-app_api_1	running	node-api-mongo-app	2024-10-19 22:55:23	172.18.0.7	3005
node-api-mongo-app_cadvisor_1	healthy	node-api-mongo-app	2024-10-19 22:55:21	172.18.0.6	8081
node-api-mongo-app_mongo_1	running	mongo	2024-10-19 22:55:21	172.18.0.4	2701
node-api-mongo-app_node-exporter_1	running	prom/node-exporter	2024-10-19 22:55:21	172.18.0.2	9100
node-api-mongo-app_portainer_1	running	portainer/portainer-ce	2024-10-19 22:55:21	172.18.0.3	9000
node-api-mongo-app_prometheus_1	running	prom/prometheus	2024-10-19 22:55:21	172.18.0.5	9090
node-api-mongo-app_web_1	running	node-api-mongo-app	2024-10-19 22:55:24	172.18.0.8	3001

The browser address bar shows the URL: 54.164.5.147:9000#!/2/docker/stacks/node-api-mongo-app?type=2&external=true. The taskbar at the bottom includes icons for React App, Portainer, Prometheus Time Series Collector, and other Microsoft and Google services.

### III. Using prom-client to monitor the Node.js Application's Metrics with Prometheus

# 24- Access client libraries at Prometheus' webpage to know how to collect the Node.js Application's Metrics with Prometheus

The screenshot shows a web browser window with the following details:

- Title Bar:** Client libraries | Prometheus
- Address Bar:** prometheus.io/docs/instrumenting/clientlibs/
- Toolbar:** Back, Forward, Stop, Refresh, Search, Home, Bookmarks, etc.
- Page Header:** Prometheus (with logo), DOCS, DOWNLOAD, COMMUNITY, SUPPORT & TRAINING, BLOG, Search, Notifications (0).
- Left Sidebar (Table of Contents):**
  - INTRODUCTION
  - CONCEPTS
  - PROMETHEUS SERVER
  - VISUALIZATION
  - INSTRUMENTING
    - Client libraries
    - Writing client libraries
    - Pushing metrics
    - Exporters and integrations
    - Writing exporters
    - Exposition formats
  - OPERATING
  - ALERTMANAGER
  - BEST PRACTICES
- Content Area:**

## CLIENT LIBRARIES

Before you can monitor your services, you need to add instrumentation to their code via one of the Prometheus client libraries. These implement the Prometheus metric types.

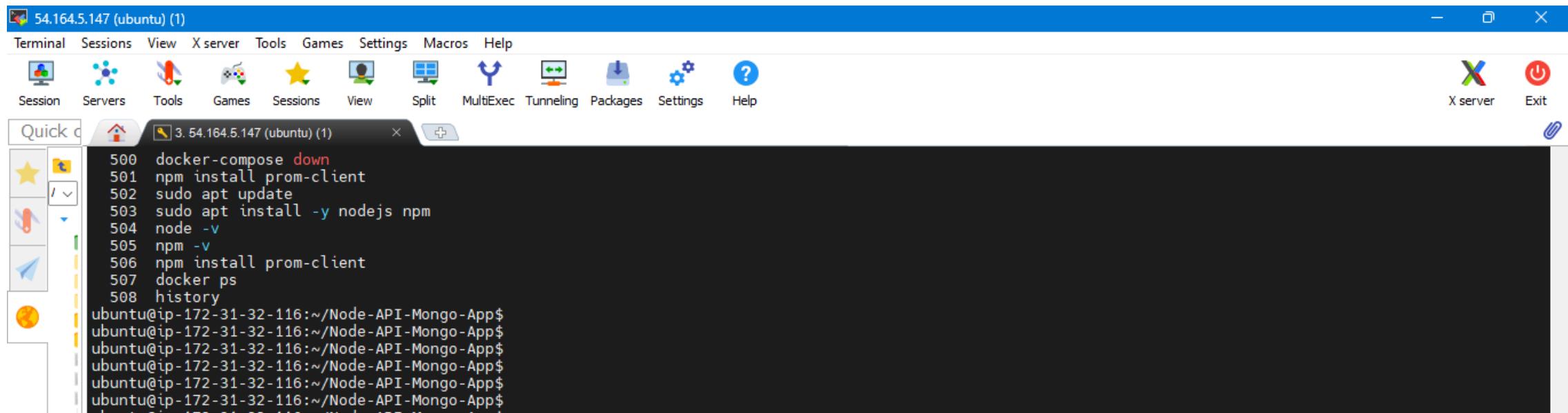
Choose a Prometheus client library that matches the language in which your application is written. This lets you define and expose internal metrics via an HTTP endpoint on your application's instance:

  - Go
  - Java or Scala
  - Python
  - Ruby
  - Rust

Unofficial third-party client libraries:

  - Bash
  - C
  - C++
  - Common Lisp
  - Dart
  - Delphi
  - Elixir
  - Erlang
  - Haskell
  - Julia
  - Lua for Nginx
  - Lua for Tarantool
  - .NET / C#
  - Node.js
  - OCaml
- System Tray/Taskbar:** Weather (23°C), Search, File Explorer, Task View, Edge, File, Mail, Google Chrome, Paint 3D, File History, 11:46 PM, ENG, Wi-Fi, Battery, 10/19/2024.

## 25- Installing prom-client to collect the Node.js Application's Metrics with Prometheus and also must update the code to the application to be able to expose the metrics at (/metrics)

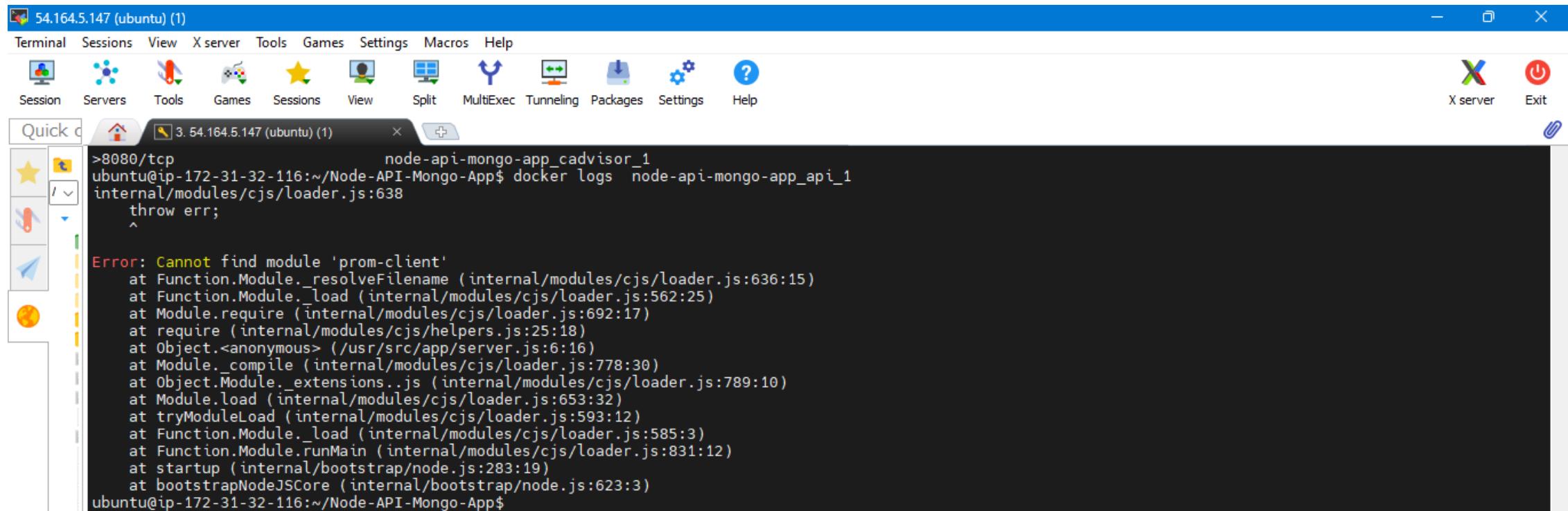


The screenshot shows a desktop environment with a terminal window open. The terminal window title is "54.164.5.147 (ubuntu) (1)". The terminal content shows the following command history:

```
500 docker-compose down
501 npm install prom-client
502 sudo apt update
503 sudo apt install -y nodejs npm
504 node -v
505 npm -v
506 npm install prom-client
507 docker ps
508 history
ubuntu@ip-172-31-32-116:~/Node-API-Mongo-App$
```

The desktop interface includes a menu bar with "Terminal", "Sessions", "View", "X server", "Tools", "Games", "Settings", "Macros", and "Help". Below the menu is a toolbar with icons for "Session", "Servers", "Tools", "Games", "Sessions", "View", "Split", "MultiExec", "Tunneling", "Packages", "Settings", and "Help". On the right side of the desktop, there are "X server" and "Exit" buttons. A sidebar on the left contains icons for "Quick c", "Session", "Servers", "Tools", "Games", "Sessions", "View", "Split", "MultiExec", "Tunneling", "Packages", "Settings", and "Help".

## 26- An error occurred and I solved it by adding prom-client library to application code



The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is "3. 54.164.5.147 (ubuntu) (1)". The terminal content shows a command being run and an error message. The command is:

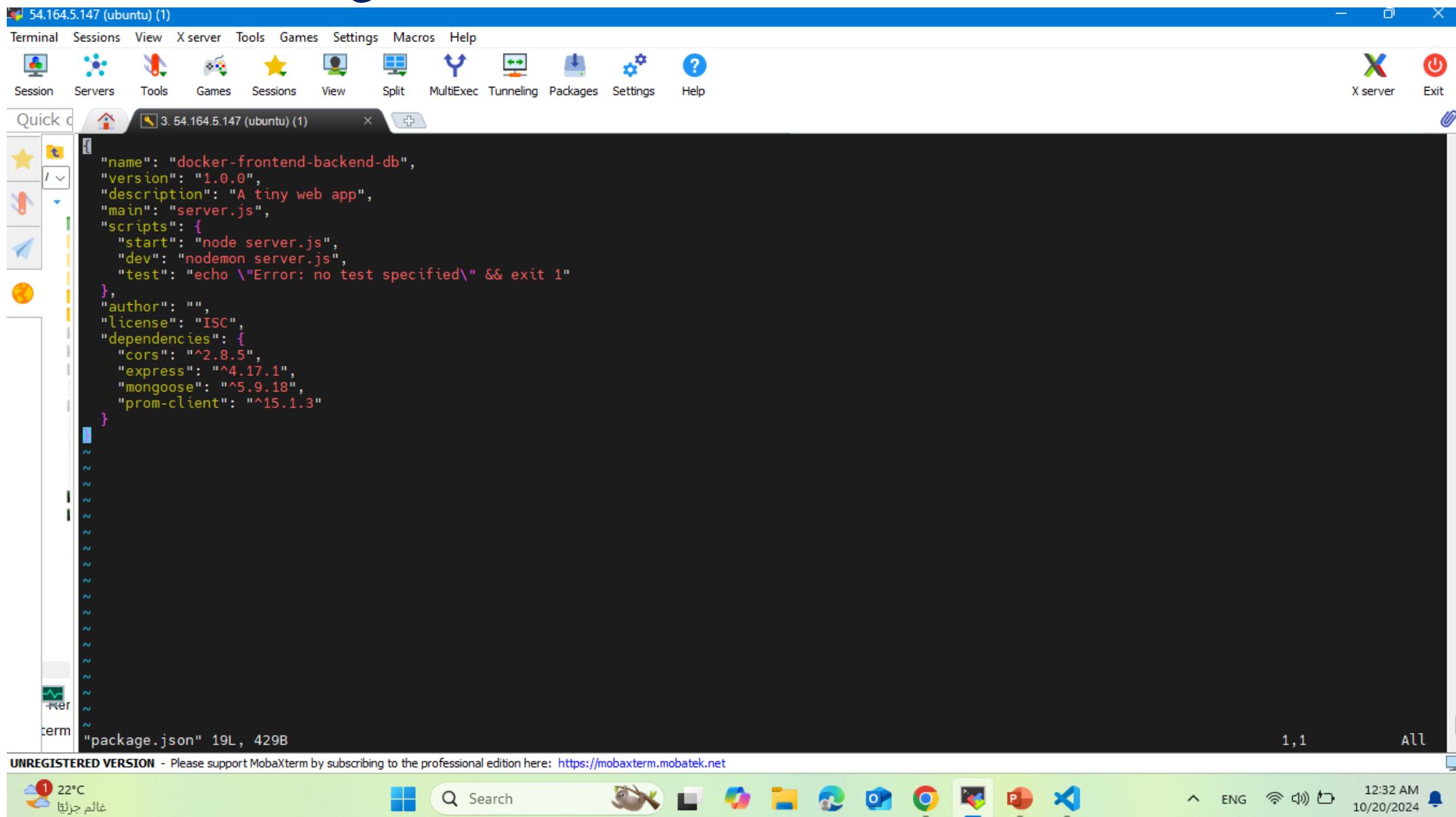
```
>8080/tcp
ubuntu@ip-172-31-32-116:~/Node-API-Mongo-App$ docker logs node-api-mongo-app_api_1
```

The error message is:

```
internal/modules/cjs/loader.js:638
    throw err;
^

Error: Cannot find module 'prom-client'
at Function.Module._resolveFilename (internal/modules/cjs/loader.js:636:15)
at Function.Module._load (internal/modules/cjs/loader.js:562:25)
at Module.require (internal/modules/cjs/loader.js:692:17)
at require (internal/modules/cjs/helpers.js:25:18)
at Object.<anonymous> (/usr/src/app/server.js:6:16)
at Module._compile (internal/modules/cjs/loader.js:778:30)
at Object.Module._extensions..js (internal/modules/cjs/loader.js:789:10)
at Module.load (internal/modules/cjs/loader.js:653:32)
at tryModuleLoad (internal/modules/cjs/loader.js:593:12)
at Function.Module._load (internal/modules/cjs/loader.js:585:3)
at Function.Module.runMain (internal/modules/cjs/loader.js:831:12)
at startup (internal/bootstrap/node.js:283:19)
at bootstrapNodeJSCore (internal/bootstrap/node.js:623:3)
ubuntu@ip-172-31-32-116:~/Node-API-Mongo-App$
```

## 27- After solving the issue



The screenshot shows a MobaXterm window titled "54.164.5.147 (ubuntu) (1)". The terminal session is displaying the contents of a file named "package.json". The JSON object contains the following data:

```
{  
  "name": "docker-frontend-backend-db",  
  "version": "1.0.0",  
  "description": "A tiny web app",  
  "main": "server.js",  
  "scripts": {  
    "start": "node server.js",  
    "dev": "nodemon server.js",  
    "test": "echo \\"$Error: no test specified\\" & exit 1"  
  },  
  "author": "",  
  "license": "ISC",  
  "dependencies": {  
    "cors": "^2.8.5",  
    "express": "^4.17.1",  
    "mongoose": "^5.9.18",  
    "prom-client": "^15.1.3"  
  }  
}
```

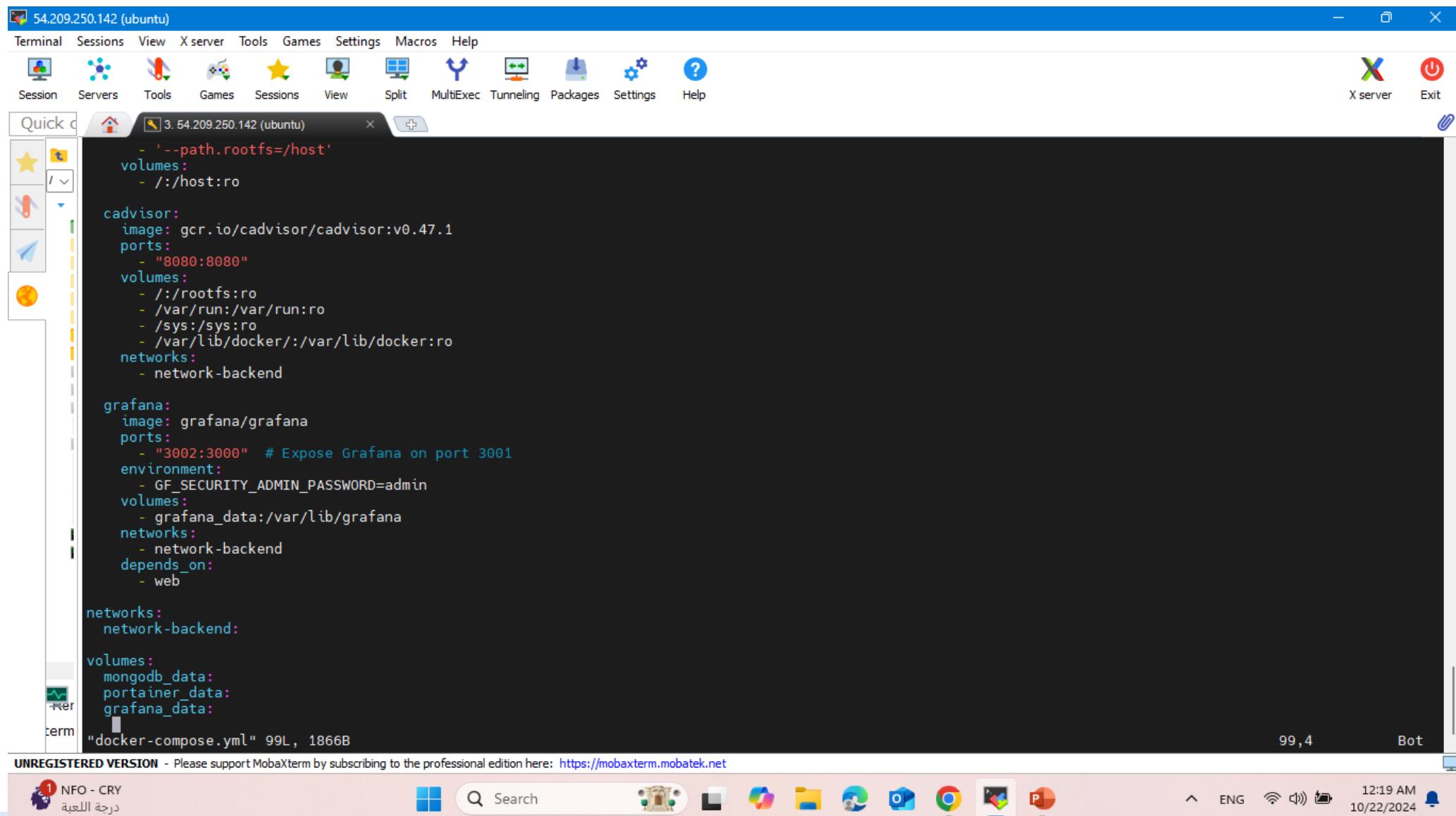
The status bar at the bottom of the terminal window indicates "package.json" 19L, 429B. The bottom right corner of the status bar shows the date and time as 10/20/2024 12:32 AM.

At the very bottom of the screen, there is a Windows taskbar with various pinned icons, including File Explorer, Microsoft Edge, and the Start button.

## Step II

# **Integrating Grafana for Virtualization**

# 28- Adding cAdvisor as a service in Docker Compose File



The screenshot shows a terminal window titled "54.209.250.142 (ubuntu)" in MobaXterm. The window displays a Docker Compose configuration file named "docker-compose.yml". The file defines three services: cAdvisor, Grafana, and a network. The cAdvisor service uses the gcr.io/cadvisor/cadvisor:v0.47.1 image and exposes port 8080. It mounts host rootfs at /host and several host system paths at /var/run, /sys, and /var/lib/docker. The Grafana service uses the grafana/grafana image and exposes port 3001. It sets GF\_SECURITY\_ADMIN\_PASSWORD=admin, mounts grafana\_data at /var/lib/grafana, and depends on the web service. The network section creates a network named "network-backend". The bottom status bar indicates the file has 99L and 1866B.

```
54.209.250.142 (ubuntu)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
X server Exit
Quick c 3. 54.209.250.142 (ubuntu) +
```
version: '3'
services:
  cadvisor:
    image: gcr.io/cadvisor/cadvisor:v0.47.1
    ports:
      - "8080:8080"
    volumes:
      - '/:/rootfs:ro'
      - '/var/run:/var/run:ro'
      - '/sys:/sys:ro'
      - '/var/lib/docker/:/var/lib/docker:ro'
    networks:
      - network-backend

  grafana:
    image: grafana/grafana
    ports:
      - "3002:3000" # Expose Grafana on port 3001
    environment:
      - GF_SECURITY_ADMIN_PASSWORD=admin
    volumes:
      - grafana_data:/var/lib/grafana
    networks:
      - network-backend
    depends_on:
      - web

  networks:
    network-backend:

volumes:
  mongodb_data:
  portainer_data:
  grafana_data:
```
"docker-compose.yml" 99L, 1866B
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net
99,4 Bot
NFO - CRY درجة اللعيبة
Search
12:19 AM 10/22/2024
ENG
Wi-Fi
Battery
P
```

# 29- All containers running in a stack

The screenshot shows the Portainer.io interface for managing Docker stacks. The browser tab is titled "Portainer | local". The address bar shows the URL: "Not secure 54.209.250.142:9000/#!/2/docker/stacks/node-api-mongo-app?type=2&external=true". The left sidebar has a dark blue theme with the following navigation items:

- Home
- local
- Dashboard
- Templates
- Stacks** (selected)
- Containers
- Images
- Networks
- Volumes
- Events
- Host

Administration section:

- User-related
- Environment-related
- Registries
- Logs
- Notifications
- Settings

At the bottom of the sidebar, it says "portainer.io Community Edition 2.21.3".

The main content area is titled "Stack details" for the stack "node-api-mongo-app". It contains two sections:

- Information:** A note stating "This stack was created outside of Portainer. Control over this stack is limited."
- Containers:** A table listing the following containers:

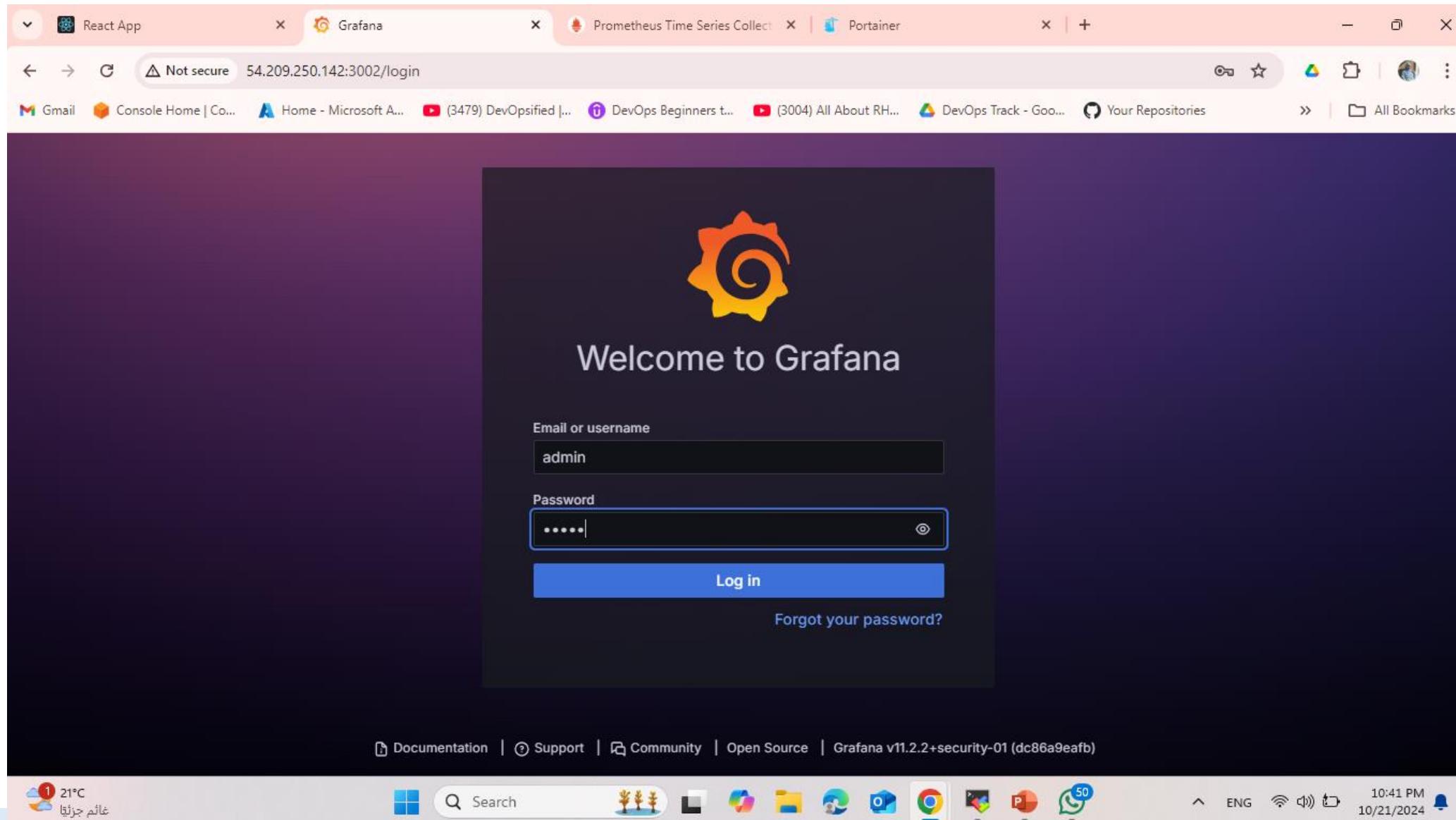
Name	State	Image	Created	IP Address	Published Ports	Ownership
node-api-mongo-app_api_1	running	node-api-mongo-app	2024-10-21 22:38:59	172.18.0.7	3001:3001	administrators
node-api-mongo-app_cadvisor_1	healthy	gcr.io/cadvisor/cadvisor:v0.47.1	2024-10-21 22:38:58	172.18.0.3	8080:8080	administrators
node-api-mongo-app_grafana_1	running	grafana/grafana	2024-10-21 22:39:01	172.18.0.9	3002:3000	administrators
node-api-mongo-app_mongo_1	running	mongo	2024-10-21 22:38:58	172.18.0.5	27017:27017	administrators
node-api-mongo-app_node-exporter_1	running	prom/node-exporter	2024-10-21 22:38:57	172.18.0.2	9100:9100	administrators
node-api-mongo-app_portainer_1	running	portainer/portainer-ce	2024-10-21 22:38:58	172.18.0.6	9000:9000	administrators
node-api-mongo-app_prometheus_1	running	prom/prometheus	2024-10-21 22:38:58	172.18.0.4	9090:9090	administrators
node-api-mongo-app_web_1	running	node-api-mongo-app	2024-10-21 22:39:00	172.18.0.8	3000:3000	administrators

At the bottom of the main content area, there is a "Items per page" dropdown set to 10.

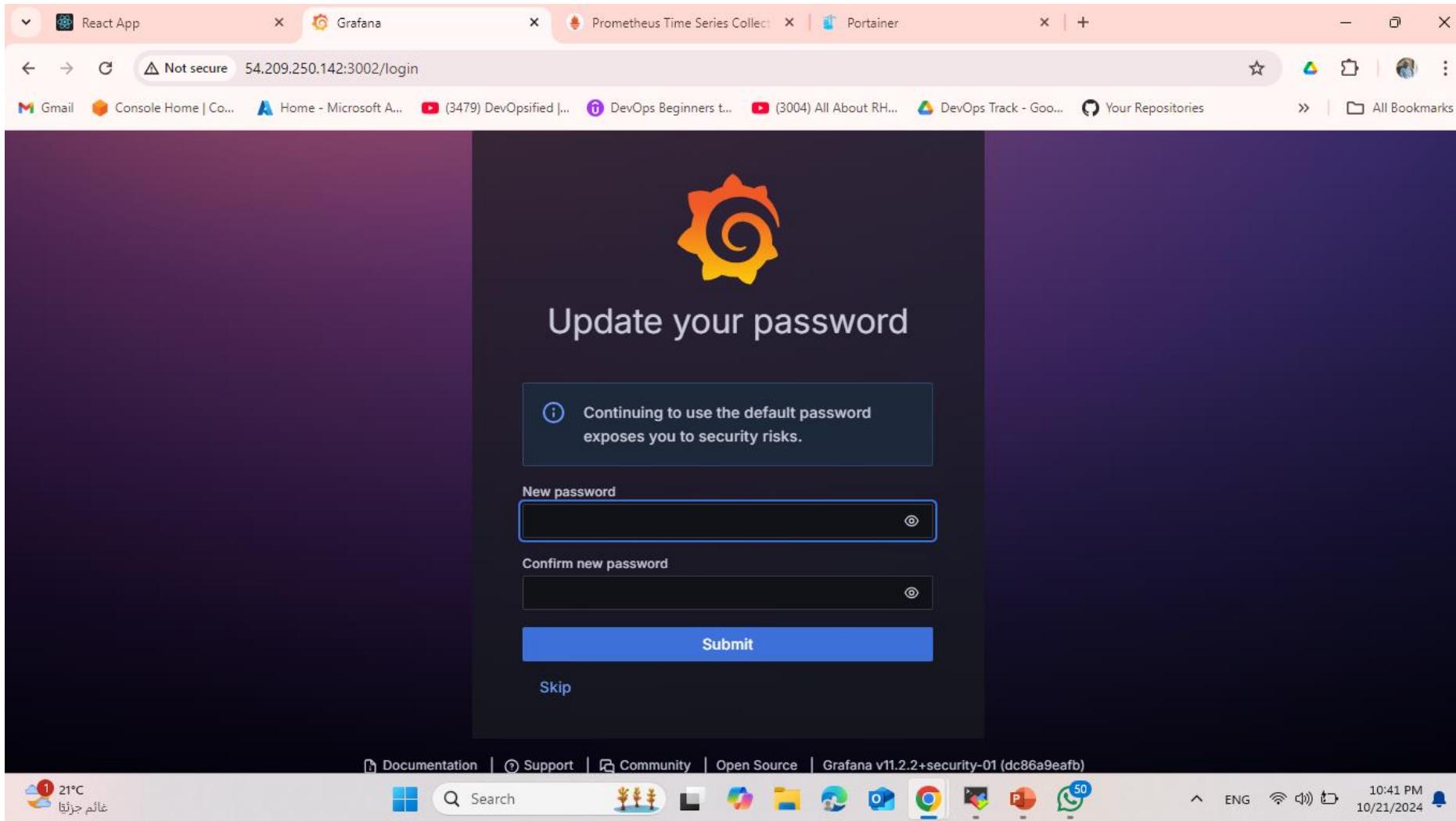
The system tray at the bottom of the screen shows the following icons and information:

- Cloud icon with "21°C" and "غامض" (Arabic)
- Windows logo
- Search icon
- File icon
- Folder icon
- Cloud storage icon
- Google Chrome icon
- Power icon
- ENG language indicator
- Wi-Fi signal
- Battery icon
- 12:26 AM timestamp
- 10/22/2024 date

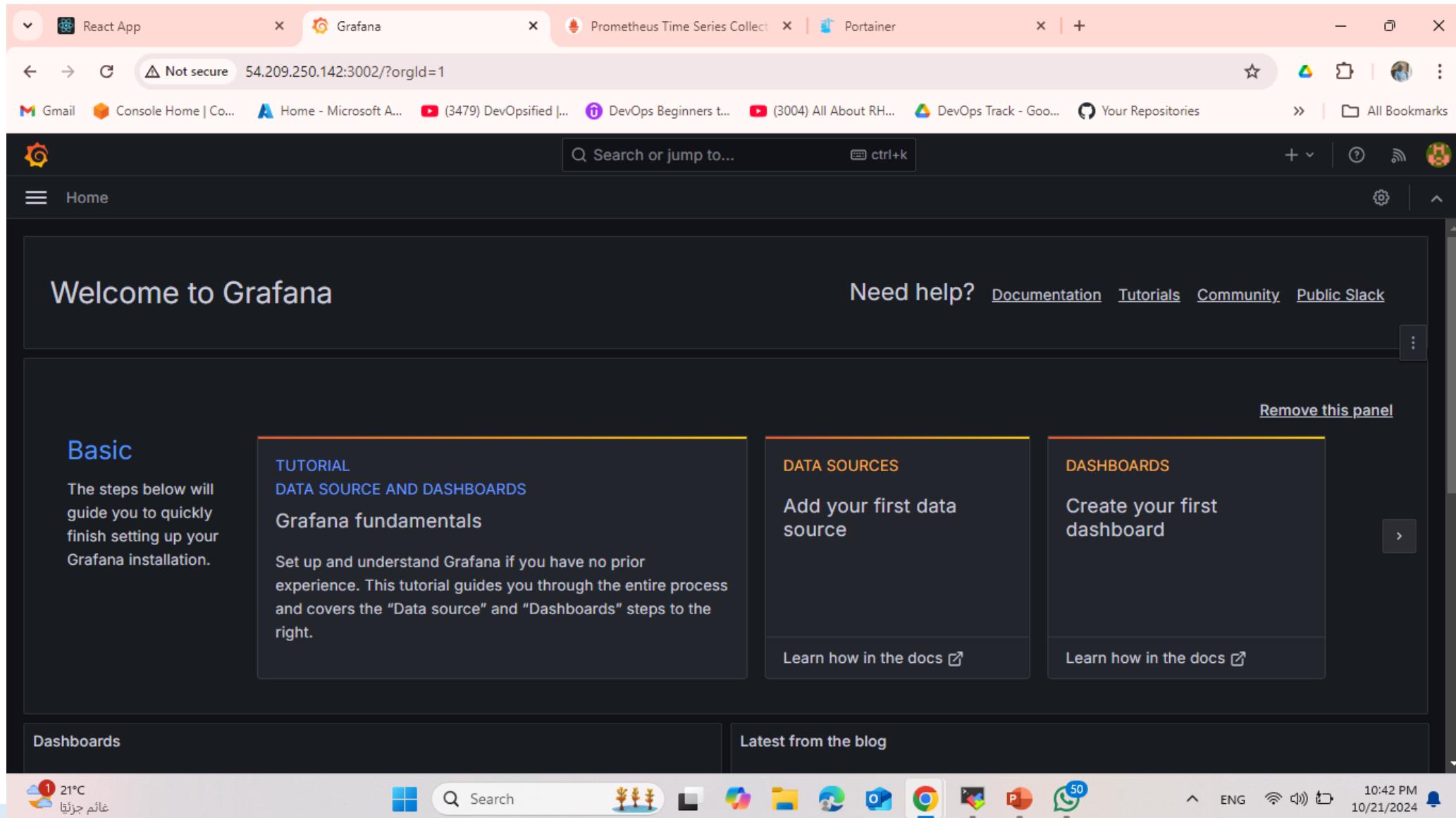
# 30- Grafana successfully running on port 3002



# 31- Updating Grafana password



## 32- Grafana gui



# 33- Adding Prometheus as data source on grafana

The screenshot shows a web browser window with the following details:

- Browser Tabs:** React App, Add data source - Data sources, Prometheus Time Series Collector, Portainer | local.
- Address Bar:** Not secure 54.209.250.142:3002/connections/datasources/new
- Toolbar:** Gmail, Console Home | Co..., Home - Microsoft A..., (3479) DevOpsified |..., DevOps Beginners t..., (3004) All About RH..., DevOps Track - Goo..., Your Repositories, All Bookmarks.
- Header:** Search or jump to... (ctrl+k), +, ?, ⚙, 🌐, 📞, 📱.
- Breadcrumbs:** Home > Connections > Data sources > Add data source
- Section:** Add data source
- Text:** Choose a data source type
- Search Bar:** Filter by name or type
- Buttons:** Cancel
- List:** Time series databases
  - Prometheus**: Open source time series database & alerting. Core.
  - Graphite**: Open source time series database. Core.
  - InfluxDB**: Open source time series database. Core.
- System Tray:** 21°C, 1, غائم جزئی, Search, 🌱, L, 🎨, 📁, 📺, 📱, 📺, 📱, 50, ENG, 11:02 PM, 10/21/2024.

# 34- Adding Prometheus as data source on grafana

The screenshot shows a web browser window with the following details:

- Tab Bar:** React App, prometheus - Data sources - C, Prometheus Time Series Collector, Portainer | local.
- Address Bar:** Not secure 54.209.250.142:3002/connections/datasources/edit/be1l2mbyh3gn4a
- Toolbar:** Search, Star, Refresh, Home, Bookmarks, etc.
- Page Header:** Home > Connections > Data sources > prometheus
- Section:** prometheus (Icon)
- Type:** Prometheus
- Status:** Alerting Supported
- Buttons:** Explore data, Build a dashboard
- Sub-section:** Type: Prometheus
- Buttons:** Settings (highlighted), Dashboards
- Input Fields:** Name (prometheus), Default (checkbox checked)
- Note:** Before you can use the Prometheus data source, you must configure it below or in the config file. For detailed instructions, [view the documentation](#).
- Text:** Fields marked with \* are required
- Section:** Connection
- Input Field:** Prometheus server URL \* (http://54.209.250.142:9090)
- System Status Bar:** 1 21°C, Search, File, Folder, Microsoft Edge, Google Chrome, Paint, WhatsApp, ENG, WiFi, 11:03 PM, 10/21/2024, Bell icon.

# 35- Adding Prometheus as data source on grafana

The screenshot shows a web browser window with multiple tabs open. The active tab is titled "prometheus - Data sources - Co..." and displays the configuration for a Prometheus data source in Grafana. The URL in the address bar is `54.209.250.142:3002/connections/datasources/edit/be1l2mbyh3gn4a`. The browser's toolbar includes icons for search, star, refresh, and various extensions.

The main content area shows the "Data sources" configuration for "prometheus". It includes sections for "Other" (Custom query parameters: Example: `max_source_resolution=5m&timeout=10s`, HTTP method: POST) and "Exemplars" (with a "+ Add" button). A success message at the bottom states: "Successfully queried the Prometheus API. Next, you can start to visualize data by building a dashboard, or by querying data in the Explore view." At the bottom of the page are "Delete" and "Save & test" buttons. The browser's status bar at the bottom right shows the date and time as 10/21/2024 11:06 PM.

# 36- Configuring Prometheus dashboard

The screenshot shows a web browser window with the address bar displaying `54.209.250.142:3002/connections/datasources/edit/be1l2mbyh3gn4a`. The page title is "prometheus - Data sources - Grafana". The browser interface includes a search bar, a tab bar with various links like "React App", "Portainer | local", and "Prometheus Time Series Collector", and a navigation bar with icons for refresh, back, forward, and search.

The main content area is a dark-themed dashboard for managing data sources. On the left, a sidebar menu is open, showing categories such as Home, Starred, Dashboards (which is selected and highlighted in blue), Playlists, Snapshots, Library panels, Public dashboards, Explore, Alerting, Connections (with "Add new connection" and "Data sources" listed under it), and Administration. The "Data sources" item is currently active, indicated by an orange border.

The central panel displays a table with one row for the "prometheus" data source. The columns are "Type" (set to "Prometheus"), "Alerting" (set to "Supported"), "Explore data", and "Build a dashboard". Below the table, there is a note: "Before you can use this data source, you must configure it below or in the config file. For detailed instructions, [view the documentation](#)".

The bottom of the screen shows the Windows taskbar with various pinned icons, including File Explorer, Google Chrome, and Microsoft Edge. The system tray shows the date and time as "10/21/2024 11:07 PM".

# 37- Configuring Prometheus dashboard

The screenshot shows a web browser window with four tabs open:

- React App
- Dashboards - Grafana
- Prometheus Time Series Collector
- Portainer | local

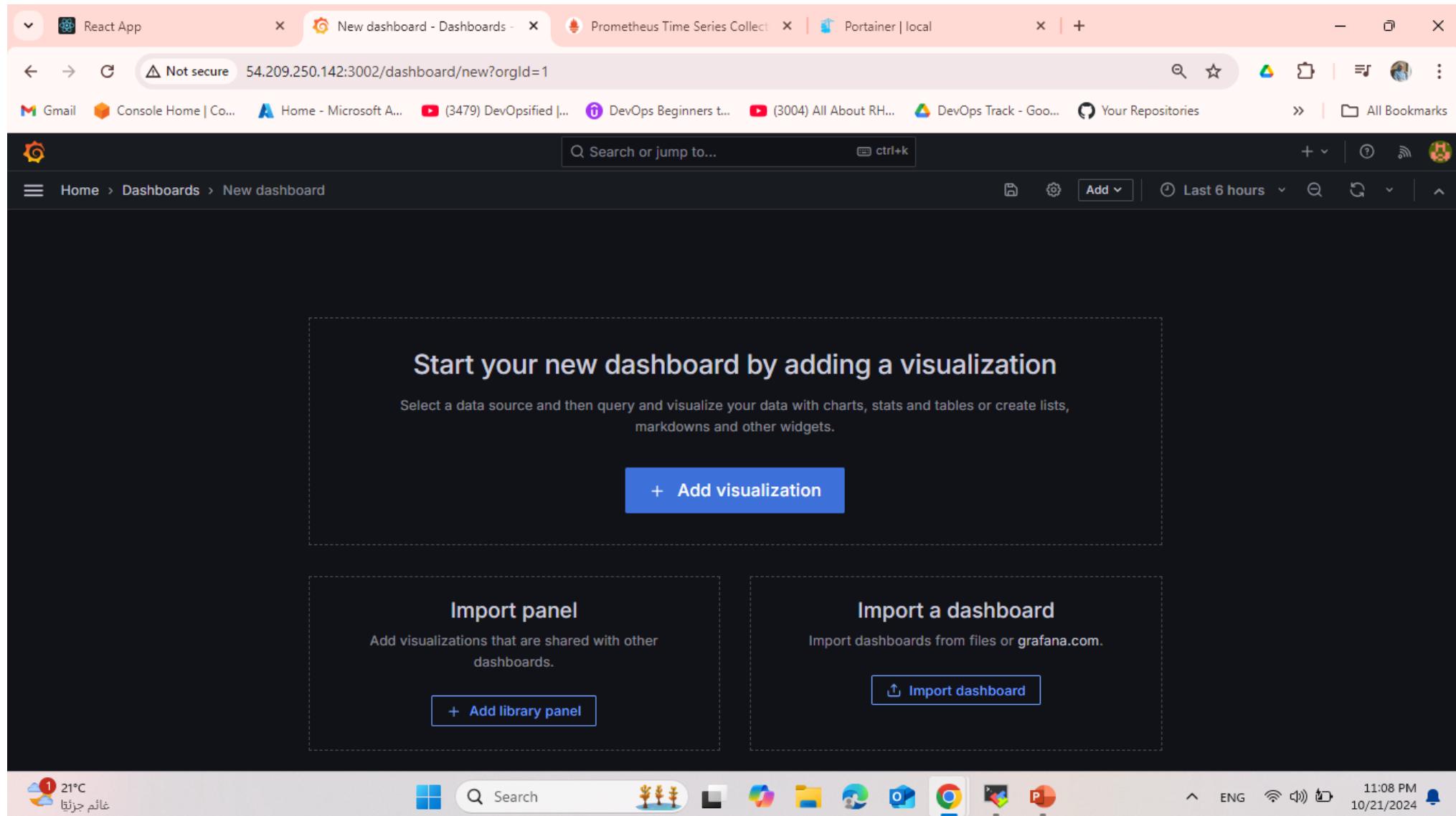
The main content area is the Grafana Dashboards interface. It features a search bar at the top labeled "Search or jump to..." with a keyboard shortcut "ctrl+k". Below the search bar is a breadcrumb navigation: "Home > Dashboards". On the right side of the header is a "New" button.

The main area is titled "Dashboards" and contains the message "Create and manage dashboards to visualize your data". It includes a search input field, a "Filter by tag" dropdown, and a "Starred" checkbox. To the right of these filters are icons for sorting and filtering.

In the center of the dashboard list area is a large, colorful cartoon sun character with a thermometer-like face, standing next to a cactus and some grass. Below the sun is the text "You haven't created any dashboards yet". At the bottom of this section is a blue button with a plus sign and the text "Create dashboard".

The bottom of the screen shows the Windows taskbar with various pinned icons (File Explorer, Google Chrome, Microsoft Edge, etc.) and system status indicators (Wi-Fi, battery, volume, date/time).

# 38- Importing node-explorer dashboard



# 39- Searching for Grafana node-explorer dashboard

The screenshot shows a Microsoft Edge browser window with the following details:

- Title Bar:** Shows the search query "grafana dashboard node export" and the URL "google.com/search?q=grafana+dashboard+node+exporter&oq=grafana+dashboard+no&gs\_lcp=EgZjaHJvbWUqbBwgAEAAgAQyBwgAEAAgAQyBwg...".
- Toolbar:** Includes standard browser controls (Back, Forward, Stop, Refresh), a search icon, and a user profile icon.
- Address Bar:** Displays the search query "grafana dashboard node exporter".
- Search Results:**
  - Sponsored:** A New Relic advertisement for "Grafana Monitoring Dashboards". It includes a link to "https://www.newrelic.com/grafana/monitoring".
  - Grafana:** A result from "Grafana Labs" titled "Node Exporter Full | Grafana Labs". It describes monitoring Linux deployment with Grafana Cloud and provides a link to "https://grafana.com/grafana/dashboards/1860-node...".
- Bottom Bar:** Shows the Windows taskbar with various pinned icons (File Explorer, Task View, Control Panel, etc.) and system status indicators (Battery, Wi-Fi, Volume, Date/Time).

# 40- Copying dashboard's id from Grafana official page

The screenshot shows a Microsoft Edge browser window with the following details:

- Title Bar:** Node Exporter Full | Grafana Lab
- Address Bar:** grafana.com/grafana/dashboards/1860-node-exporter-full/
- Toolbar:** Includes links to Gmail, Console Home, Home - Microsoft, DevOps Beginners, DevOps Track, Your Repositories, and All Bookmarks.
- Grafana Labs Header:** Products, Open Source, Solutions, Learn, Docs, Company, Downloads, Contact us, Sign in.
- Dashboard Preview:** Shows two panels: one with CPU/Memory/Net/Disk metrics and another with Memory/Meminfo metrics.
- Description:** Nearly all default values exported by Prometheus node exporter graphed. Only requires the default job\_name: node, add as many targets as you need in /etc/prometheus/prometheus.yml.
- Configuration Snippet:**

```
- job_name: node
  static_configs:
    - targets: ['localhost:9100']
```

A "Copy" button is available next to this snippet.
- Get this dashboard:** A sidebar with two steps:
  - Sign up for Grafana Cloud [Create free account](#)
  - Import the dashboard template [Copy ID to clipboard](#) or [Download JSON](#)
- System Tray:** Shows weather (21°C), search bar, file explorer, taskbar icons (Cloud, File, Mail, Google Chrome, Photos, Powerpoint), and system status (ENG, 11:09 PM, 10/21/2024).

# 41- Copying dashboard's id to grafana to be imported

The screenshot shows a web browser window with multiple tabs open. The active tab is titled "Import dashboard - Dashboard" and has the URL "54.209.250.142:3002/dashboard/import". The browser's address bar also displays "Not secure 54.209.250.142:3002/dashboard/import". The page content is the "Import dashboard" interface, which includes a file upload area for JSON files, a search bar, and a text input field containing the value "1860". Below this is a code editor showing a partial JSON model for a dashboard.

**Import dashboard**

Import dashboard from file or [Grafana.com](#)

Upload dashboard JSON file

Drag and drop here or click to browse  
Accepted file types: .json, .txt

Find and import dashboards for common applications at [grafana.com/dashboards](#)

1860

Import via dashboard JSON model

```
{  
  "title": "Example - Repeating Dictionary variables",  
  "uid": "_OHnEoN4z",  
  "panels": [...]  
}
```

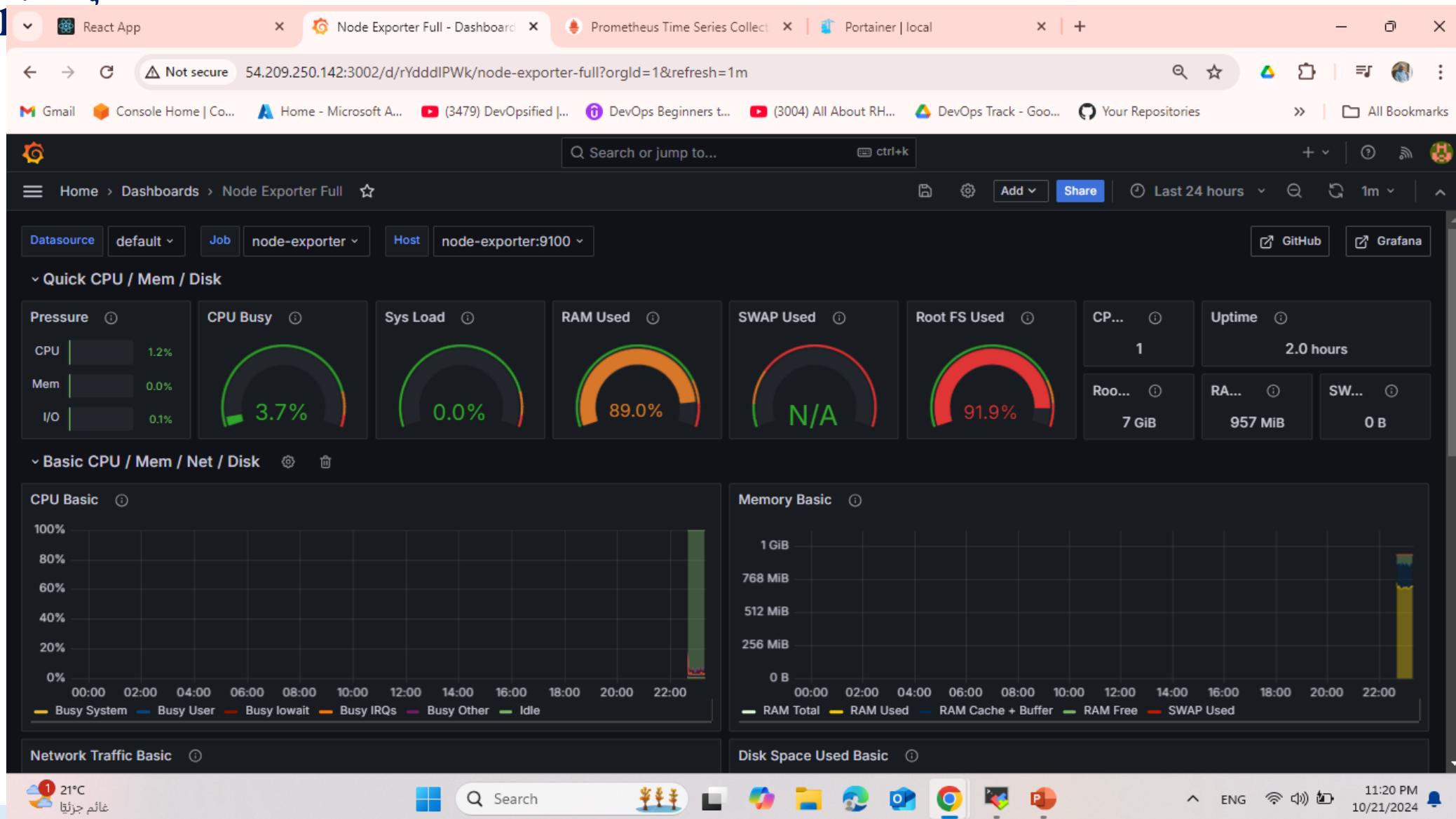
21°C 10/21/2024 11:10 PM ENG

# 42- Copying dashboard's id to grafana to be imported

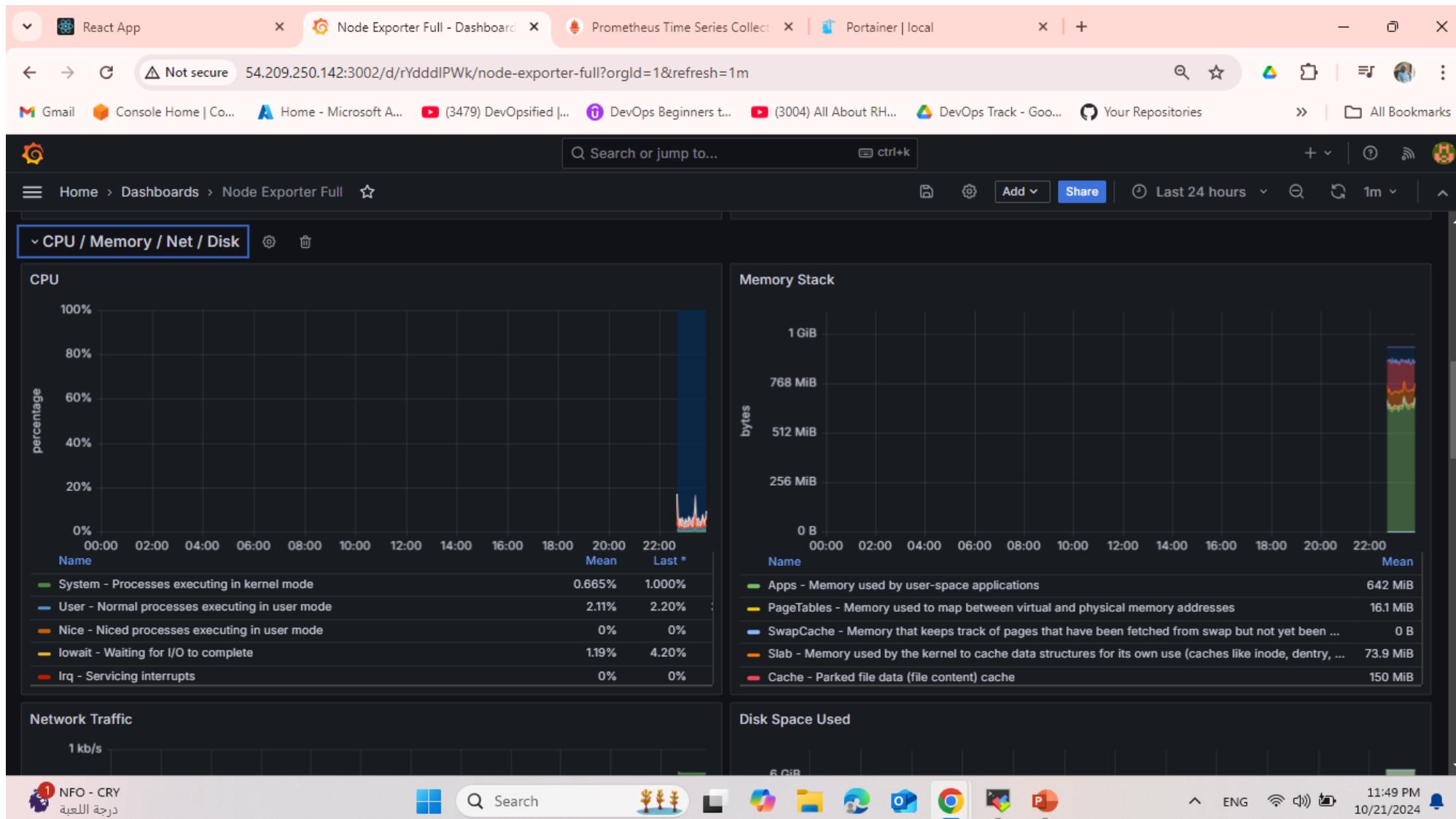
The screenshot shows a browser window with the following details:

- Tab Bar:** React App, Import dashboard - Dashboard, Prometheus Time Series Collector, Portainer | local.
- Address Bar:** Not secure 54.209.250.142:3002/dashboard/import
- Toolbar:** Back, Forward, Stop, Refresh, Search, Favorites, Home, Bookmarks, etc.
- Content Area:**
  - Import dashboard:** Title and subtext "Import dashboard from file or Grafana.com".
  - Importing dashboard from Grafana.com:** Subtitle.
  - Published by:** rfmoz
  - Updated on:** 2024-05-22 16:07:35
  - Options:**
    - Name:** Node Exporter Full
    - Folder:** Dashboards
  - Unique identifier (UID):** A detailed description explaining that the UID allows for consistent URLs even if the dashboard title changes. It includes a current UID value (rYdddIPWk) and a "Change uid" button.
  - Prometheus:** A section related to Prometheus monitoring.
- Bottom Bar:** Taskbar with icons for File Explorer, Control Panel, Task View, File, OneDrive, Google Chrome, and Powerpoint.
- System Tray:** Shows battery level (21%), language (ENG), signal strength, and date/time (10/21/2024, 11:10 PM).

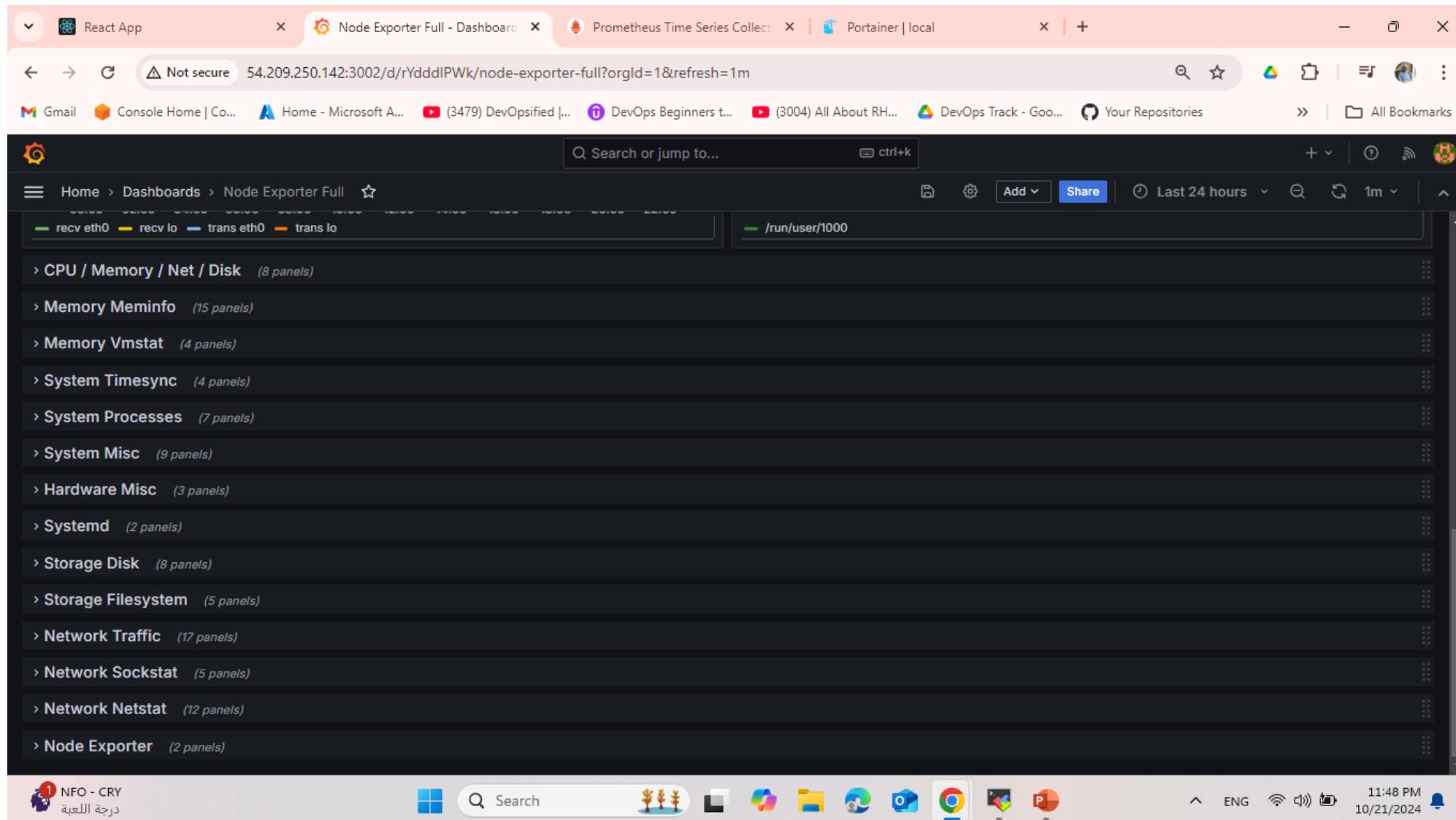
# 43- Dashboard successfully imported and is viewing host metrics



# 44- Dashboard successfully imported and is viewing host metrics'



# 45- Dashboard successfully imported and is viewing host metrics'



# 46- Creating a new dashboard to visualize containers' metrics through cAdvisor

The screenshot shows a web browser window with the following details:

- Browser Tabs:** React App, Dashboards - Grafana, Prometheus Time Series Collector, Portainer | local.
- Address Bar:** Not secure 54.209.250.142:3002/dashboards
- Toolbar:** Gmail, Console Home | Co..., Home - Microsoft A..., (3479) DevOpsified |..., DevOps Beginners t..., (3004) All About RH..., DevOps Track - Goo..., Your Repositories, All Bookmarks.
- Content Area:** The main content is the Grafana Dashboards page. It features a search bar ("Search or jump to..."), filter options ("Filter by tag", "Starred"), and a list of dashboards. One dashboard, "Node Exporter Full", is listed under the "Name" filter. To the right, there's a "Tags" section with a single tag labeled "linux".
- Context Menu:** A context menu is open on the right side of the screen, with the "New dashboard" option highlighted. Other options in the menu include "New folder" and "Import".
- Address Bar (Bottom):** 54.209.250.142:3002/dashboard/import
- System Tray:** Shows icons for user profile, weather (21°C), and system status.
- Taskbar:** Shows various application icons including File Explorer, Edge, and others.
- System Status:** Top right corner shows ENG, battery level, signal strength, and the date/time (10/21/2024, 11:50 PM).

# 47- Searching for Grafana cAdvisor dashboard

The screenshot shows a Google search results page for the query "grafana dashboard cAdvisor". The results are displayed in a standard Google search layout with three main entries:

- Cadvisor exporter | Grafana Labs**  
Get this **dashboard**. 1. Sign up for **Grafana** Cloud. Create free account. 2. Import the **dashboard** template.
- cAdvisor dashboard | Grafana Labs**  
I created this **cAdvisor dashboard** to show everything scraped from **cAdvisor**. Expanding all the rows may consume too much resources. Limit the containers by ...
- cAdvisor Docker Insights | Grafana Labs**  
**cAdvisor** Docker Insights. This **Grafana dashboard** offers a basic overview of key performance metrics for Docker containers in your system.

The browser interface includes a toolbar at the top with various icons and a status bar at the bottom showing the date and time.

# 48- Copying cadvisor's dashboard's id from Grafana official page

The screenshot shows a web browser window with the title "cadvisor dashboard | Grafana". The URL in the address bar is "grafana.com/grafana/dashboards/19792-cadvisor-dashboard/". The main content area displays the "cadvisor dashboard" with multiple panels showing system metrics and logs. On the right side, a dark-themed modal window titled "Get this dashboard" is open. It contains two numbered steps: 1. "Sign up for Grafana Cloud" with a "Create free account" button, and 2. "Import the dashboard template" with a "Dashboard ID copied!" message and a "Download JSON" button. The browser's status bar at the bottom shows the date and time as "10/21/2024 11:59 PM".

# 49- Copying cadvisor dashboard's id to grafana to be imported

The screenshot shows a browser window with several tabs open at the top:

- React App
- Import dashboard - Dashboard
- Prometheus Time Series Collect
- Portainer | local

The main content area is titled "Import dashboard" and contains the following sections:

- Upload dashboard JSON file**: A dashed box with an upward arrow icon, a "Load" button, and a text input field containing "19792".
- Find and import dashboards for common applications at [grafana.com/dashboards](https://grafana.com/dashboards)**
- Import via dashboard JSON model**: A code editor showing a JSON snippet:

```
{  
  "title": "Example - Repeating Dictionary variables",  
  "uid": "_0HnEoN4z",  
  "panels": [...]  
}
```

The browser's address bar shows the URL: `54.209.250.142:3002/dashboard/import`. The status bar at the bottom right shows the date and time: `10/21/2024 11:51 PM`.

# 50- Copying dashboard's id to grafana to be imported

Import dashboard

Import dashboard from file or Grafana.com

Importing dashboard from [Grafana.com](#)

Published by [simonmysun](#)

Updated on 2024-05-23 14:10:59

Options

Name

Folder

Unique identifier (UID)  
The unique identifier (UID) of a dashboard can be used for uniquely identify a dashboard between multiple Grafana installs. The UID allows having consistent URLs for accessing dashboards so changing the title of a dashboard will not break any bookmarked links to that dashboard.

9AJV\_mnlk [Change uid](#)

Prometheus

React App Import dashboard - Dashboard Prometheus Time Series Collector Portainer | local

1 21°C ENG 11:52 PM 10/21/2024

# 51- Configuring prometheus

The screenshot shows a web browser window with multiple tabs open. The active tab is titled "Import dashboard - Dashboard" and has the URL [54.209.250.142:3002/dashboard/import](https://54.209.250.142:3002/dashboard/import). The browser interface includes a search bar, a toolbar with various icons, and a bookmarks bar at the bottom.

The main content area displays a "Importing dashboard from Grafana.com" page. It shows details about the dashboard being imported:

- Published by: simonmysun
- Updated on: 2024-05-23 14:10:59

Below this, there are "Options" for configuring the imported dashboard:

- Name:** cadvisor dashboard
- Folder:** Dashboards
- Unique identifier (UID):** 9AJV\_mnlk (with a "Change uid" button)
- Prometheus:** prometheus (selected)

At the bottom of the form are two buttons: "Import" (highlighted in blue) and "Cancel".

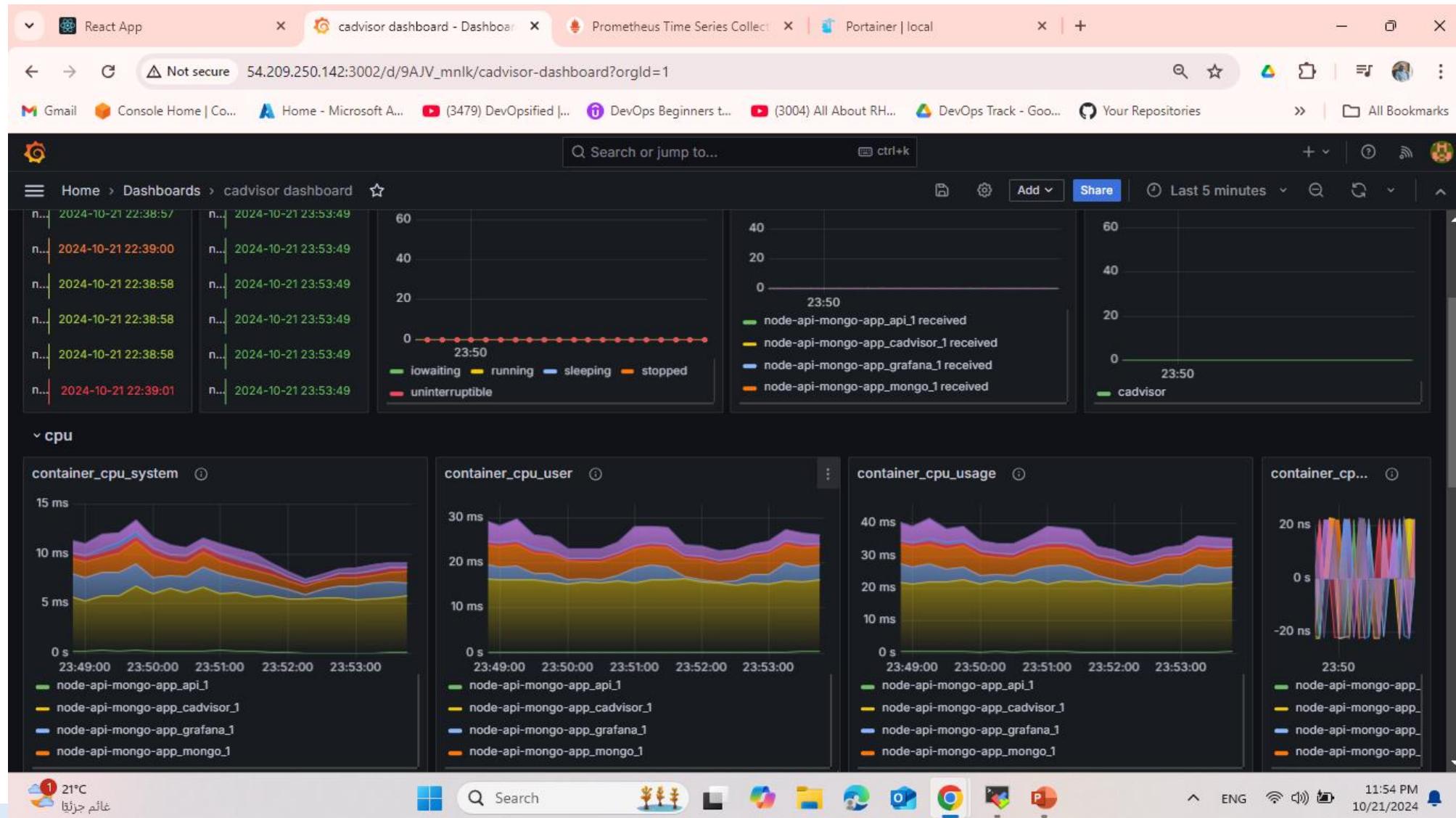
The browser's taskbar at the bottom shows several pinned icons, including Microsoft Edge, File Explorer, and other productivity tools. The system tray indicates the date and time as 10/21/2024 at 11:52 PM.

# 52- Dashboard successfully imported and is viewing all containers' metrics'

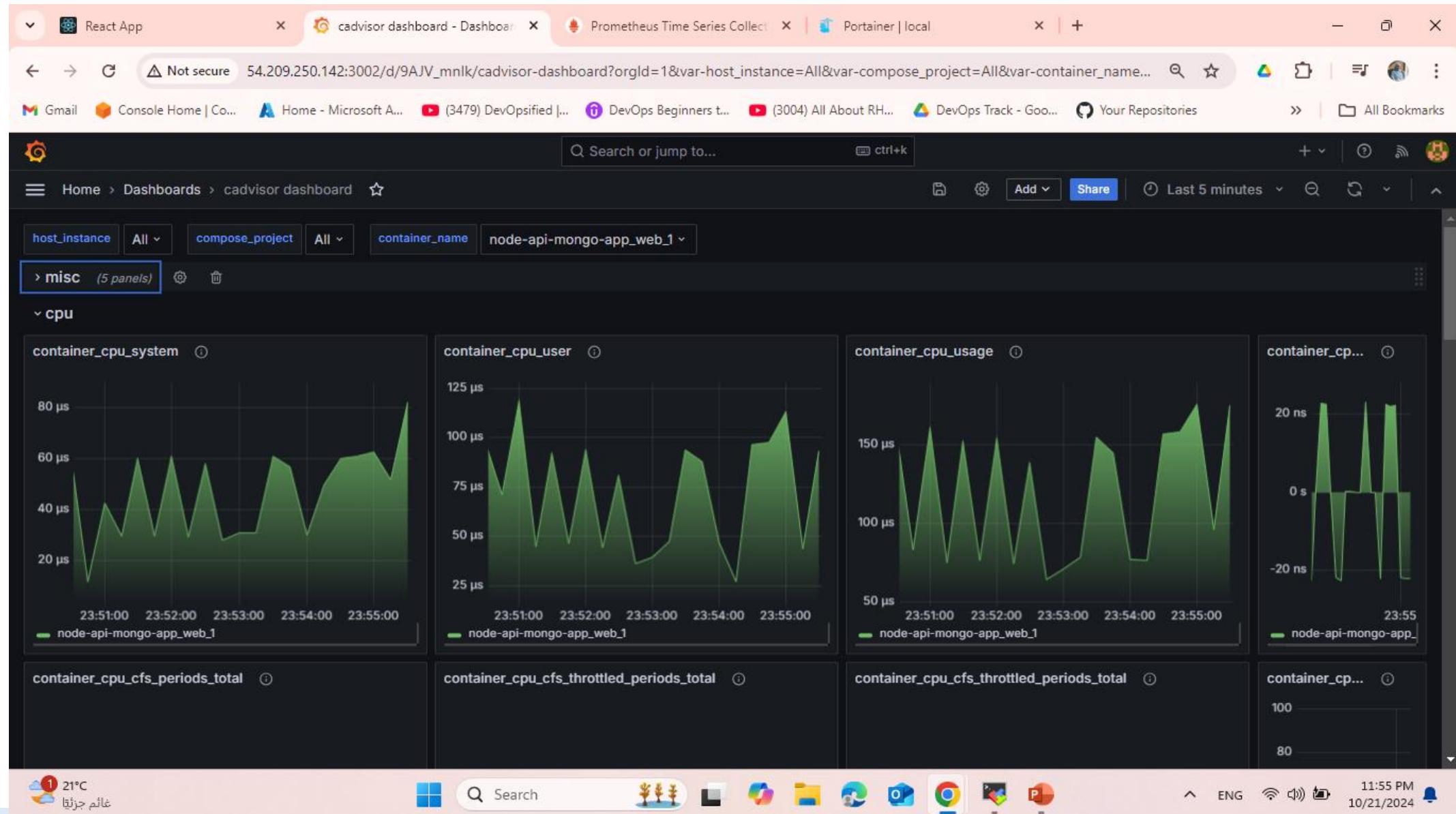
The screenshot shows a web browser window with the following details:

- Browser Tabs:** React App, cAdvisor dashboard - Dashboard (highlighted), Prometheus Time Series Collect, Portainer | local.
- Address Bar:** Not secure 54.209.250.142:3002/d/9AJV\_mnlk/cadvisor-dashboard?orgId=1
- Search Bar:** Search or jump to... (ctrl+k)
- Sidebar:** Home > Dashboards > cAdvisor dashboard
- Filter Options:** host\_instance (All), compose\_project (All), container\_name (Enter variable value). A dropdown menu is open under 'container\_name' with the following items:
  - Selected (1)
  - All
  - node-api-mongo-app\_api\_1
  - node-api-mongo-app\_cadvisor\_1
  - node-api-mongo-app\_grafana\_1
  - node-api-mongo-app\_mongo\_1
  - node-api-mongo-app\_node-exporter\_1
  - node-api-mongo-app\_portainer\_1
  - node-api-mongo-app\_prometheus\_1
  - node-api-mongo-app\_web\_1
- Content Area:** Lists categories like misc, cpu, memory, network, blkio, fs, and various container nodes (e.g., Container node-api-mongo-app\_api\_1, Container node-api-mongo-app\_cadvisor\_1).
- Taskbar:** Shows system icons for weather (21°C), search, file explorer, task manager, and other applications.
- Bottom Right:** System tray showing ENG, battery level, and date/time (11:53 PM, 10/21/2024).

# 53- Dashboard successfully imported and is viewing all containers' metrics'



# 54- Any container's metrics could be viewed separately



# 55- The two dashboards

The screenshot shows a web browser window with four tabs open:

- React App
- Dashboards - Grafana
- Prometheus Time Series Collector
- Portainer | local

The "Dashboards - Grafana" tab is active, displaying the Grafana interface. The address bar shows the URL: `54.209.250.142:3002/dashboards`. The page title is "Dashboards".

Key elements of the interface include:

- A search bar at the top labeled "Search or jump to...".
- A "New" button in the top right corner.
- A sidebar on the left with a "Home" icon and the path "Home > Dashboards".
- A main content area titled "Dashboards" with the sub-instruction "Create and manage dashboards to visualize your data".
- A search bar labeled "Search for dashboards and folders".
- Filtering options: "Filter by tag" dropdown, "Starred" checkbox, and "Sort" dropdown.
- A table listing dashboards:

Name	Tags
cadvisor dashboard	linux
Node Exporter Full	

- A system tray at the bottom showing icons for weather (21°C), search, file explorer, task manager, and browser.
- A status bar at the bottom right showing the date (10/21/2024), time (11:56 PM), and battery level.

# Github Repository

<https://github.com/salmasalam024/Node-API-Mongo-App>