**📘 TASK 7: Monitor System Resources Using Netdata**

**🎯 Objective:**

To install **Netdata**, a real-time performance monitoring tool, using **Docker** and to visualize system and container metrics like CPU usage, memory, disk I/O, and network activity.

**🧰 Tools Used:**

* Docker (installed on local system)
* Netdata (official image from Docker Hub)
* Web browser (to access the dashboard)

**📋 Step-by-Step Installation and Monitoring Process:**

**✅ Step 1: Check if Docker is Installed and Running**

Open your terminal or Command Prompt and run:

bash

CopyEdit

docker --version

🎯 **Purpose**: Verifies Docker is correctly installed.

Expected Output (example):

nginx

CopyEdit

Docker version 24.0.5, build ce2d111

**✅ Step 2: Pull the Netdata Image from Docker Hub**

Run the command:

bash

CopyEdit

docker pull netdata/netdata

🎯 **Purpose**: Downloads the official Netdata image so Docker can run it as a container.

Expected Output:

vbnet

CopyEdit

Using default tag: latest

latest: Pulling from netdata/netdata

...

Status: Downloaded newer image for netdata/netdata:latest

**✅ Step 3: Run the Netdata Container**

bash

CopyEdit

docker run -d --name=netdata -p 19999:19999 netdata/netdata

📌 **Explanation of each flag:**

| **Flag** | **Explanation** |
| --- | --- |
| -d | Run container in background (detached mode) |
| --name=netdata | Assign a name for the container |
| -p 19999:19999 | Map port 19999 on container to same port on host |
| netdata/netdata | The image to run (Netdata official) |

Expected Output:

scss

CopyEdit

d9081282db596e43241348b550e4b51f... (Container ID)

**✅ Step 4: Access Netdata in the Browser**

Open your browser and visit:

arduino

CopyEdit

http://localhost:19999

🎯 **Purpose**: Launches the Netdata web interface to monitor live system metrics.

🟢 You should now see a **dashboard** like the one below with charts for:

* CPU
* RAM
* Disk I/O
* Network Traffic
* Docker containers (if any running)

📷 Screenshot proof:  
*(Attach your screenshot here)*

**✅ Step 5: Verify the Container is Running**

Run:

bash

CopyEdit

docker ps

Expected Output:

bash

CopyEdit

CONTAINER ID IMAGE PORTS NAMES

d9081282db59 netdata/netdata 0.0.0.0:19999->19999/tcp netdata

🎯 **Purpose**: Ensures Netdata is still running in the background.

**✅ Step 6: Monitor Metrics in Dashboard**

In the dashboard, explore:

* **Avg CPU per Node**: Real-time CPU usage
* **Avg Used RAM per Node**: Live memory utilization
* **Disk I/O**: Reads and writes
* **Network**: Inbound and outbound bandwidth
* **System Load**: Average load over time

💡 You can also explore:

* Alerts (predefined triggers)
* Log files
* Docker container performance

**✅ Step 7: (Optional) View Logs**

To access container logs:

bash

CopyEdit

docker exec -it netdata bash

cd /var/log/netdata

ls

🎯 **Purpose**: View internal logs like error logs, health checks, etc.

Example log files:

* error.log
* access.log

**✅ Step 8: Stop and Remove the Container (Optional)**

If you want to stop Netdata:

bash

CopyEdit

docker stop netdata

To remove the container completely:

bash

CopyEdit

docker rm netdata

**📌 Conclusion:**

Netdata was successfully installed using Docker and configured to monitor the system in real time. The dashboard provides powerful and interactive charts that display CPU usage, memory load, disk activity, network traffic, and more.

**Screenshot:** (Attached above)

**📖 1️⃣ What does Netdata monitor?**

**Netdata** is a real-time health monitoring and performance troubleshooting tool for systems and applications.  
It monitors:

* **CPU usage**
* **RAM / memory usage**
* **Disk I/O**
* **Network traffic**
* **Process stats**
* **System load**
* **Web servers (like Nginx, Apache)**
* **Databases (like MySQL, PostgreSQL)**
* **Docker containers, Kubernetes pods**
* Custom apps via plugins/collectors

It gives **per-second** (or even sub-second) granularity, unlike other tools that collect data every few seconds/minutes.

**📊 2️⃣ How do you view real-time metrics?**

**Real-time metrics** in Netdata are displayed through:

* A **web-based interactive dashboard** (usually available at http://<server-ip>:19999)
* It auto-updates in real-time (without refreshing)
* Each metric has **interactive charts** showing per-second data streams
* You can drill down to a specific process, disk, or network interface

**⚖️ 3️⃣ How is Netdata different from Prometheus?**

| **📌 Feature** | **Netdata** | **Prometheus** |
| --- | --- | --- |
| **Focus** | Real-time, high-frequency, system & app health monitoring | Time-series data collection for long-term storage & analysis |
| **Granularity** | Per second/sub-second | Typically 15s/30s/60s scrape intervals |
| **Storage** | Local (or short-term retention) | Long-term time-series DB |
| **Alerting** | Built-in, simple rules-based | Requires Alertmanager |
| **Installation** | Single-node agent (lightweight) | Requires server + exporters |
| **Use case** | Troubleshooting live incidents | Monitoring and alerting over time |

They actually work well **together** — many setups use **Netdata for instant visibility** and **Prometheus for historical analytics**.

**📥 4️⃣ What is a collector?**

A **collector** in Netdata is:

* A small plugin or module that gathers data from a source (like CPU stats, Docker stats, or MySQL metrics)
* It runs on your system and feeds metrics into Netdata
* Netdata includes 100+ built-in collectors for different services, or you can create custom ones

Example:

* proc.plugin collects Linux /proc system data
* mysql.chart.py collects MySQL performance metrics

**📈 5️⃣ What are some performance KPIs to watch?**

Key Performance Indicators (KPIs) in system monitoring could include:

* **CPU utilization (%)**
* **Load average**
* **Memory usage (used, free, cached)**
* **Disk I/O (read/write MBps, IOPS, latency)**
* **Network traffic (incoming/outgoing bandwidth, packet errors/drops)**
* **Response time (for web servers/databases)**
* **Service availability**
* **Docker container health & resource usage**

Netdata visualizes all these in real-time with alerts when thresholds are crossed.

**🖥️ 6️⃣ How to deploy Netdata on a VM?**

Simple steps:

bash

CopyEdit

bash <(curl -Ss https://my-netdata.io/kickstart.sh)

This one-liner installs the latest Netdata agent.

Then, open your browser at:

cpp

CopyEdit

http://<vm-ip>:19999

You’ll see the dashboard immediately.  
You can also configure Netdata Cloud to view multiple VMs/servers from a single UI.

**🚨 7️⃣ How does Netdata alerting work?**

Netdata uses **simple, rule-based alerts**:

* Alerts are defined in config files (health.d directory)
* Conditions are set using thresholds and durations (e.g., cpu\_usage > 90% for 10 seconds)
* Supports multiple notification methods:
  + Email
  + Slack
  + Discord
  + PagerDuty
  + Custom scripts
* Alerts trigger when conditions are met and resolve when metrics return to normal

You can customize or disable alerts per chart.

**📊 8️⃣ What is a dashboard in this context?**

In Netdata:

* A **dashboard** is a **real-time web interface** displaying system/application metrics via charts and graphs
* It’s interactive — you can zoom into time ranges, group metrics by source (CPU, disk, Docker, etc.), and customize views
* You can access the dashboard:
  + On the local agent (http://localhost:19999)
  + Or via **Netdata Cloud** to unify multiple nodes into one central view