

# **GUÍA DE INSTALACIÓN Y CONFIGURACIÓN DE UN ENTORNO DE MONITORIZACIÓN CON MUNIN**



**Nombre: José Alberto**

**Apellidos: Hoces Castro**

**DNI:**

**3º Doble Grado en Ingeniería Informática y Matemáticas**

**Ingeniería de Servidores**

# ÍNDICE

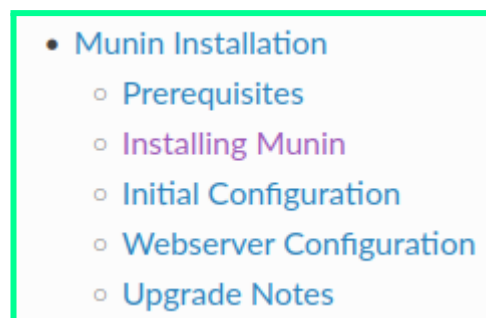
- **1. Instalación**
  - 1.1. Instalación del servidor en Ubuntu
  - 1.2. Instalación de nodos en Red Hat
- **2. Configuración**
  - 2.1. Configuración del servidor en Ubuntu
  - 2.2. Configuración de nodos en Red Hat
- **3. Monitorización con Munin**
- **4. Referencias**

# 1. Instalación

Nuestro primer paso en el proceso de instalación es entrar en la página de Munin, cuyo enlace web es: <http://munin-monitoring.org>. Una vez dentro de la página web, en la parte superior de esta se nos despliegan las siguientes opciones:



De entre todas las opciones, seleccionamos “Munin Guide”. Una vez dentro de la guía de Munin, buscamos la sección “Munin Installation”:



Y seleccionamos “Installing Munin”. Nosotros procederemos a realizar la instalación por paquetes. Esta es la más recomendada ya que la distribución de Munin por paquetes es la más testeada y la que más usan los desarrolladores. Dentro de “Installing Munin” buscamos el sistema operativo de nuestras máquinas (en mi caso Ubuntu para el anfitrión y Red Hat para la MV de la cual voy a obtener métricas).

## 1.1. Instalación del servidor en Ubuntu

### Debian/Ubuntu

Munin is distributed with both Debian and Ubuntu.

In order to get Munin up and running type

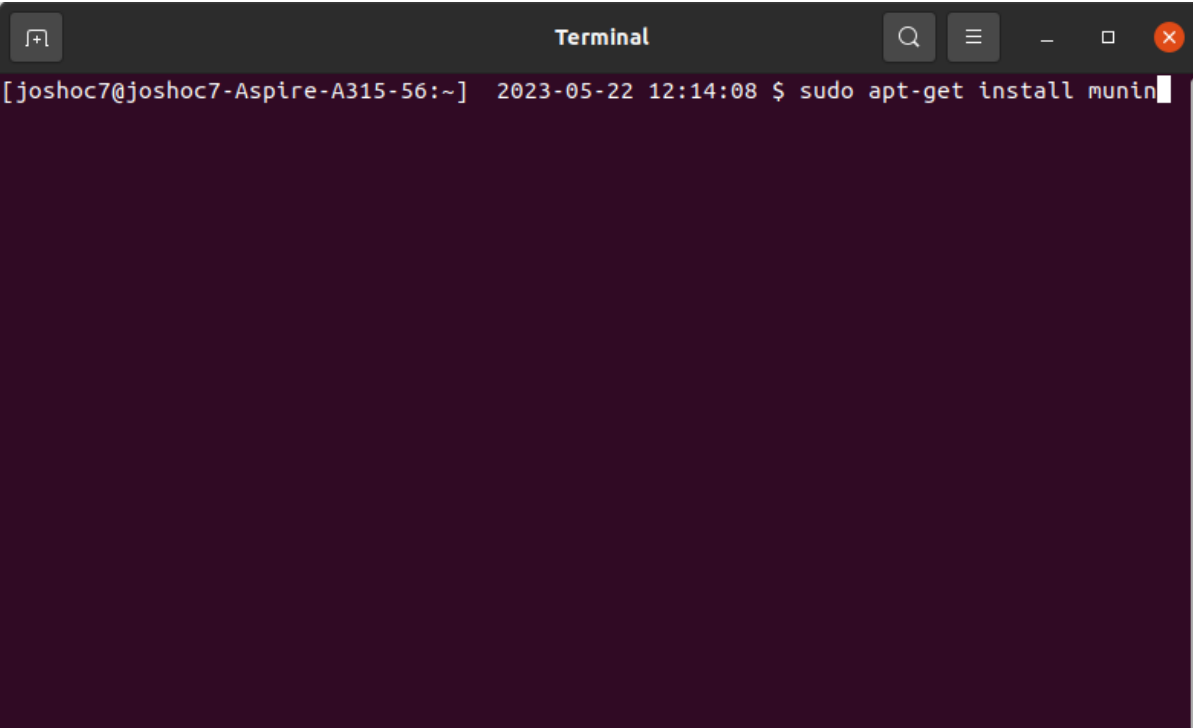
```
sudo apt-get install munin-node
```

on all nodes, and

```
sudo apt-get install munin
```

on the master.

En la sección “Installing Munin”, se nos ofrecen estos dos comandos para la instalación de Munin en Ubuntu. Debemos saber que en este caso, el comando que nos interesa es el segundo, pues *master* se refiere a la máquina que va a obtener las métricas de las demás, mientras que *node* se refiere a las máquinas que van a ser monitorizadas. Por ello, abrimos una terminal en Ubuntu, copiamos y pegamos `sudo apt-get install munin` y lo ejecutamos para que se realice la instalación:



```
Terminal
[joshoc7@joshoc7-Aspire-A315-56:~] 2023-05-22 12:14:08 $ sudo apt-get install munin
```

## 1.2. Instalación de nodos en Red Hat

Para poder instalar munin-node, es necesario instalar epel-release en la máquina Red Hat:

This article will help you to install Munin Network Monitoring Tool on your CentOS, Fedora, and Red Hat systems.

### Step 1 – Enable Yum Repository

First of all, you need to add enable EPEL yum repository in your system. Use the following command to install as per system architecture.

```
# yum install epel-release
```

### Step 2 – Install Munin Packages

En nuestra terminal de Red Hat , ejecutamos *sudo yum install epel-release*. Una vez instalado, volvemos a la guía de instalación de Munin:

#### RedHat / CentOS / Fedora

Current versions are available at [EPEL](#).

In order to install Munin type

```
sudo yum install munin-node
```

on all nodes, and

```
sudo yum install munin
```

on the master.

You will have to enable the services in systemd to get them up and running.

Como ahora estamos trabajando con la máquina virtual que vamos a monitorizar, esta va a ser un *node*. Por ello, copiaremos y ejecutaremos el comando *sudo yum install munin-node* en la terminal de nuestra máquina Red Hat:

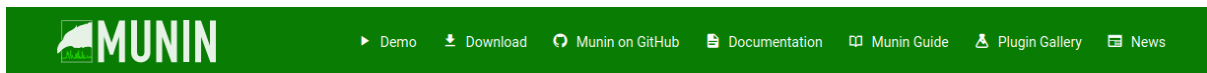
```
[jahoces@localhost ~]$ sudo yum install munin-node
```



Seleccionamos en todo lo que se nos pregunte yes “y”. Una vez tenemos instalado el paquete *munin-node* en Red Hat, procedemos a la configuración mediante la modificación de ciertos archivos que se explican en el siguiente epígrafe.

## 2. Configuración

Volvemos a la página inicial con el enlace <http://munin-monitoring.org/>. En la barra de la que antes hablábamos, seleccionamos “Download”:



Una vez dentro de dicha sección, bajamos hasta encontrar la sección “Configuration”:

### Configuration

#### On each node

The munin node listens on all interfaces by default, but has a restrictive access list. You need to add your master's IP address.

```
# /etc/munin/munin-node.conf
# A list of addresses that are allowed to connect.
# regular expression, since Net::Server does not use
# network notation unless the perl module Net::CIDR
# may repeat the allow line as many times as you'd
allow '^127.'
allow '^192.0.2.1$'
# If you have installed the Net::CIDR perl module,
# cidr_allow and cidr_deny address/mask patterns. A
# match any cidr_allow, and not match any cidr_deny
```

Start munin-node service, or restart it if it was already started:

```
user@node01:~$ sudo service munin-node restart
[ ok ] Stopping Munin-Node: done.
[ ok ] Starting Munin-Node: done.
```

#### On the master

Add the previously configured nodes to the master configuration file:

```
# /etc/munin/munin.conf
# a simple host tree
[node01.example.com]
    address 192.0.2.4

[node02.example.com]
    address node02.example.com

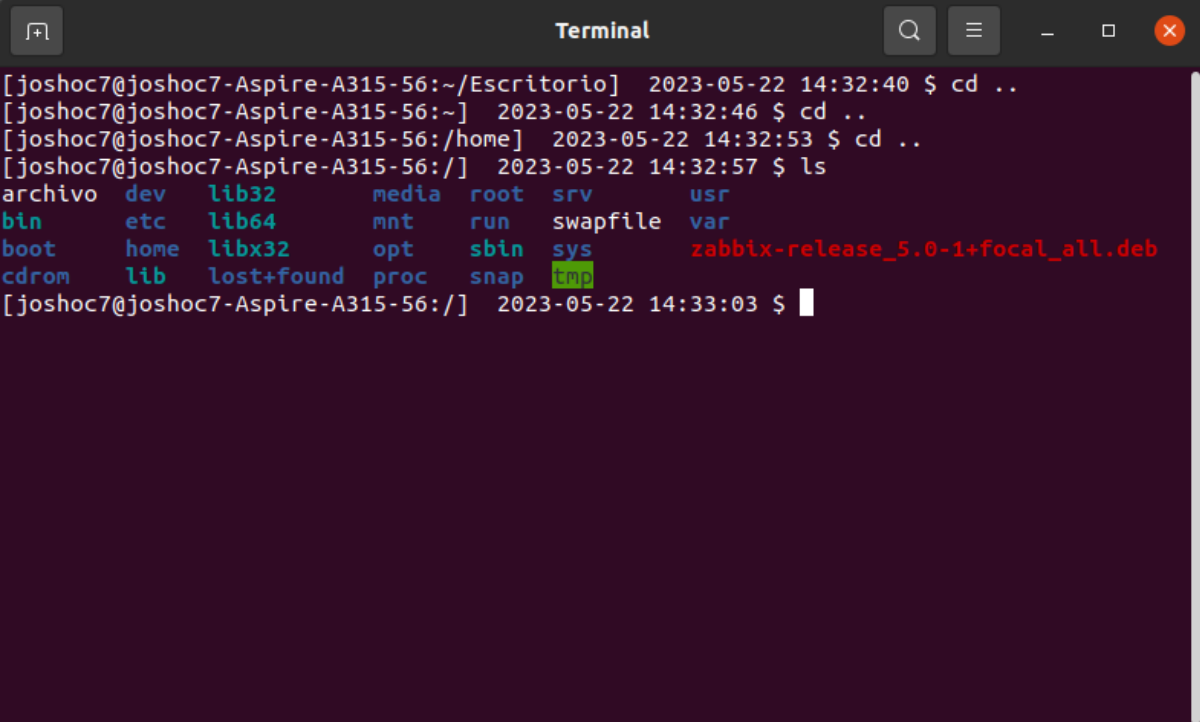
[node03.example.com]
    address 2001:db8::de:caf:bad
#
# A more complex example of a host tree
#
## First our "normal" host.
# [fii.foo.com]
```

The master service does not need to be reloaded: its configuration file will be read on the next poll.

Aquí se nos dice que el siguiente paso es modificar los archivos de configuración *munin-node.conf* en el nodo Red Hat y *munin.conf* en el servidor Ubuntu. Como podemos ver en la captura, ambos archivos se localizan en el directorio */etc*. Procedamos a buscarlos y modificarlos:

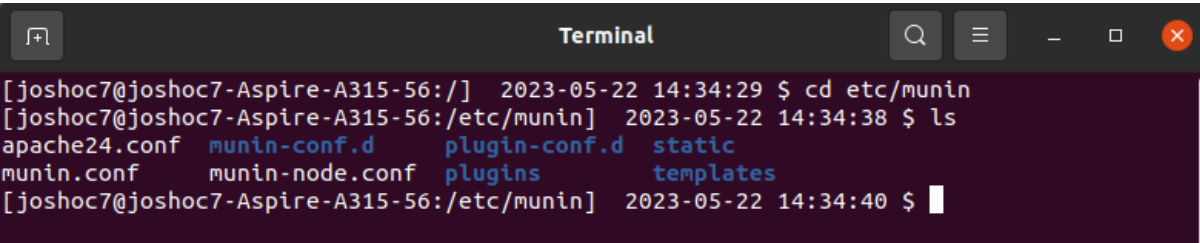
## 2.1. Configuración del servidor Ubuntu

Desde el directorio de nuestro escritorio, debemos salir al directorio de nuestro usuario, de este al directorio home, y de este último al directorio raíz /:



```
[joshoc7@joshoc7-Aspire-A315-56:~/Escritorio] 2023-05-22 14:32:40 $ cd ..
[joshoc7@joshoc7-Aspire-A315-56:~] 2023-05-22 14:32:46 $ cd ..
[joshoc7@joshoc7-Aspire-A315-56:/home] 2023-05-22 14:32:53 $ cd ..
[joshoc7@joshoc7-Aspire-A315-56:/] 2023-05-22 14:32:57 $ ls
archivo  dev    lib32  media  root  srv    usr
bin      etc    lib64  mnt    run   swapfile  var
boot     home  libx32  opt    sbin  sys      zabbix-release_5.0-1+focal_all.deb
cdrom    lib   lost+found  proc  snap  tmp
[joshoc7@joshoc7-Aspire-A315-56:/] 2023-05-22 14:33:03 $
```

Al hacer *ls*, vemos que ya tenemos a nuestro alcance el directorio */etc*. A continuación nos adentramos en el directorio */etc* y concretamente en el subdirectorio */munin* con el comando *cd etc/munin*:



```
[joshoc7@joshoc7-Aspire-A315-56:/] 2023-05-22 14:34:29 $ cd etc/munin
[joshoc7@joshoc7-Aspire-A315-56:/etc/munin] 2023-05-22 14:34:38 $ ls
apache24.conf  munin-conf.d  plugin-conf.d  static
munin.conf     munin-node.conf  plugins        templates
[joshoc7@joshoc7-Aspire-A315-56:/etc/munin] 2023-05-22 14:34:40 $
```

Y para editar *munin.conf* usaremos el editor *vi*. La finalidad de editar este archivo es añadir los nodos que se quieren monitorizar. Para ello, ejecutamos *sudo vi munin.conf* y escribimos en el archivo:



```
# a simple host tree
[localhost.localdomain]
  address 127.0.0.1
  use_node_name yes

[redhatnode]
  address 192.168.56.110
```

Recordar que para editar el archivo, primero hay que pulsar la tecla de la letra “i” para poder escribir. Una vez escrito lo que queremos, se pulsa Esc y se escribe :wq y se le da a Intro.

De esta forma, hemos configurado nuestro servidor para monitorizar un nodo cuya dirección IP es la de nuestra máquina Red Hat (192.168.56.110). Finalmente, se restablece el servicio para que se guarden los cambios con el comando `sudo systemctl restart munin`.

## 2.2. Configuración de nodos en Red Hat

El proceso para situarse en el directorio `/etc/munin` es análogo al visto en Ubuntu:

```
[jahoces@localhost ~]$ cd ..
[jahoces@localhost home]$ cd ..
[jahoces@localhost ~]$ ls
afs boot etc lib media opt root sbin sys usr
bin dev home lib64 mnt proc run srv tmp var
[jahoces@localhost ~]$ cd etc/munin/
[jahoces@localhost munin]$ ls
munin-node.conf plugin-conf.d plugins
[jahoces@localhost munin]$ _
```

En este caso necesitamos editar el archivo `munin-node.conf` ya que necesitamos permitir que el nodo escuche desde la dirección del servidor, que en este caso es la 192.168.56.1. Ejecutamos el comando `sudo vi munin-node.conf`:

```

# Set this if the client doesn't report the correct hostname when
# telnetting to localhost, port 4949
#
#host_name localhost.localdomain

# A list of addresses that are allowed to connect. This must be a
# regular expression, since Net::Server does not understand CIDR-style
# network notation unless the perl module Net::CIDR is installed. You
# may repeat the allow line as many times as you'd like

allow ^127\.0\.0\.1$
allow ^::1$
allow ^192\.168\.56\.1$

# If you have installed the Net::CIDR perl module, you can use one or more
# cidr_allow and cidr_deny address/mask patterns. A connecting client must
# match any cidr_allow, and not match any cidr_deny. Note that a netmask
# *must* be provided, even if it's /32
#
# Example:
#
#cidr_allow 127.0.0.1/32
#cidr_allow 192.0.2.0/24
#cidr_deny 192.0.2.42/32

# Which address to bind to;
host *
#host 127.0.0.1
host_name redhatnode

# And which port
port 4949

"munin-node.conf" 67L, 1582B written

```

Como se ve en la captura, se añaden dos líneas:

- Una línea `allow ^192\.168\.56\.1$` (cada uno debe poner su IP correspondiente y con el formato indicado en *munin-node.conf* como en este caso).
- Una línea `host_name redhatnode` (así, cuando monitoricemos desde el servidor Ubuntu, el nodo Red Hat aparecerá en la sección *Groups* con el nombre *redhatnode*).

Guardamos con `:wq`. El último paso que queda por dar es habilitar el puerto 4949 en el firewall de Red Hat (4949 es el puerto por defecto que usa Munin):

```

[jahoces@localhost munin]# sudo firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp0s3 enp0s8
  sources:
  services: cockpit dhcpv6-client ssh
  ports: 22022/tcp
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
[jahoces@localhost munin]# sudo firewall-cmd --add-port 4949/tcp --permanent
success
[jahoces@localhost munin]# sudo firewall-cmd --reload
success
[jahoces@localhost munin]# sudo firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp0s3 enp0s8
  sources:
  services: cockpit dhcpv6-client ssh
  ports: 22022/tcp 4949/tcp
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
[jahoces@localhost munin]#

```

Finalmente, procedemos a restablecer el servicio *munin-node* y comprobar que se queda en estado *activo*. Para ello ejecutamos en la máquina de Red Hat los comandos `sudo systemctl restart munin-node` y `sudo systemctl status munin-node`:

```

[jahoces@localhost munin]# sudo systemctl restart munin-node
[sudo] password for jahoces:
[jahoces@localhost munin]# sudo systemctl status munin-node
● munin-node.service - Munin Node
   Loaded: loaded (/usr/lib/systemd/system/munin-node.service; disabled; vendor preset: disabled)
   Active: active (running) since Mon 2023-05-22 15:04:55 CEST; 10s ago
     Docs: man:munin-node(1)
           http://guide.munin-monitoring.org/en/latest/node/index.html
   Main PID: 4088 (munin-node)
     Tasks: 1 (limit: 5896)
    Memory: 9.3M
       CPU: 544ms
   CGroup: /system.slice/munin-node.service
           └─4088 /usr/bin/perl -wT /usr/sbin/munin-node --foreground

May 22 15:04:54 localhost.localdomain systemd[1]: Starting Munin Node...
May 22 15:04:55 localhost.localdomain systemd[1]: Started Munin Node.
[jahoces@localhost munin]# _

```

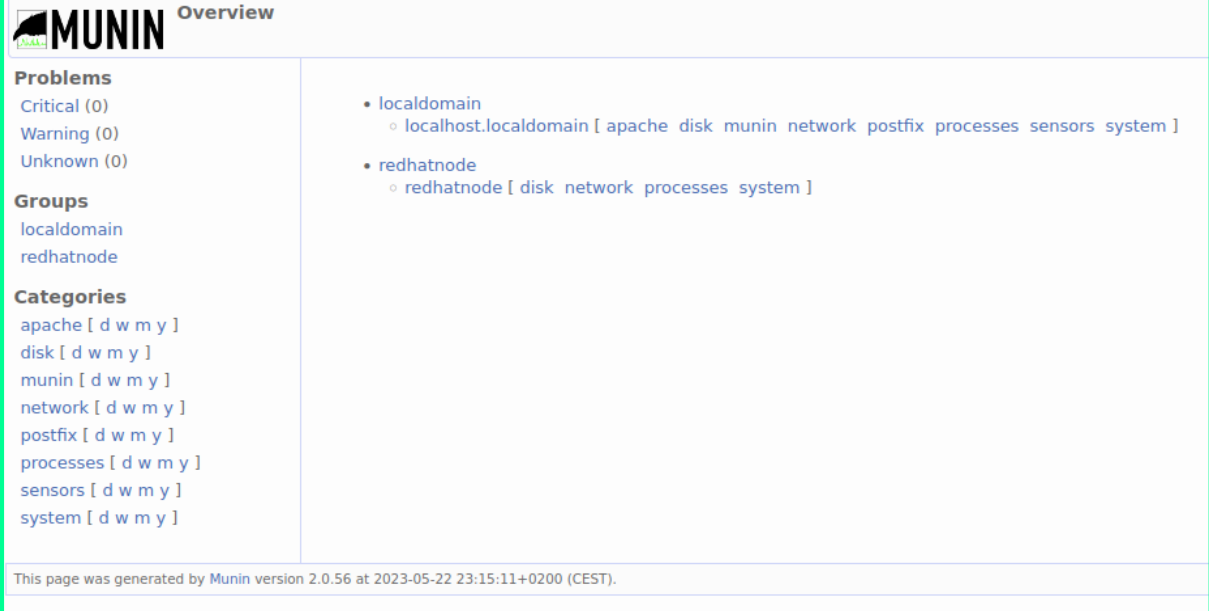
Para comprobar que la conexión con Munin funciona, ejecutamos el comando `nc 192.168.56.110 4949` desde una terminal del servidor:

```
[joshoc7@joshoc7-Aspire-A315-56:/etc/munin] 2023-05-22 23:13:13 $ nc 192.168.56.110 4949
# munin node at redhatnode
```

Se nos responde con `# munin node at redhatnode`, lo cual quiere decir que la configuración se ha realizado con éxito.

### 3. Monitorización con Munin

Para comenzar a monitorizar desde el anfitrión Ubuntu y obtener las métricas que deseamos, escribimos en nuestro buscador la URL <http://localhost/munin/>. Se nos presenta la siguiente interfaz:



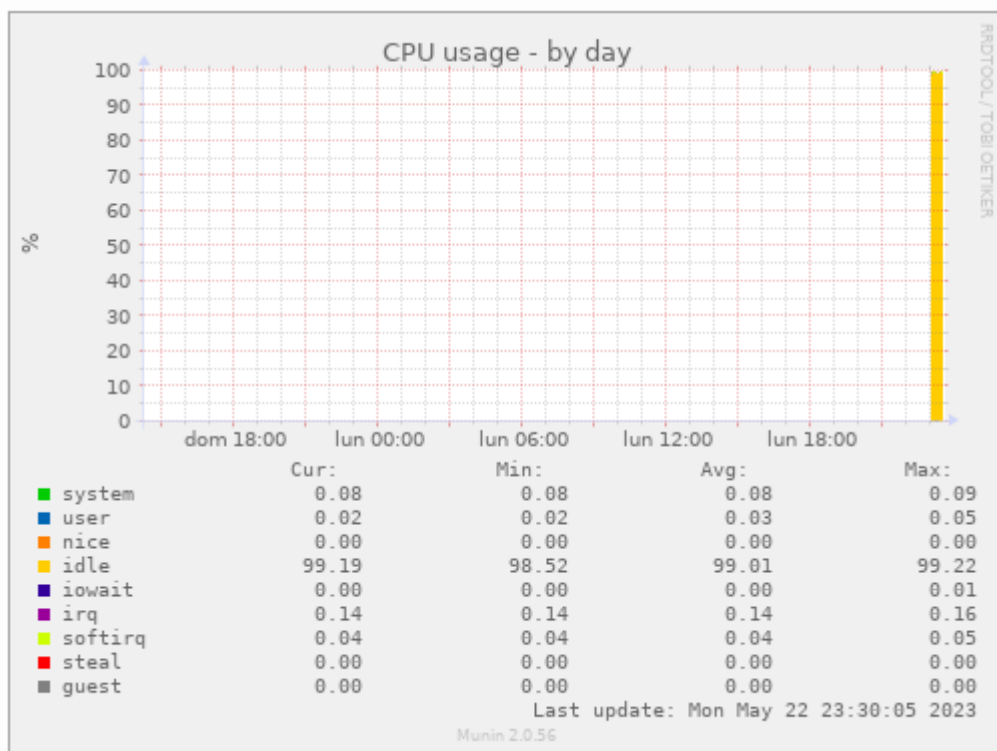
The screenshot shows the Munin Overview page. On the left, there are sections for 'Problems' (Critical, Warning, Unknown), 'Groups' (localdomain, redhatnode), and 'Categories' (apache, disk, munin, network, postfix, processes, sensors, system). The main content area on the right lists the monitored nodes: 'localdomain' and 'redhatnode', each with a list of metrics being monitored. At the bottom, a footer indicates the page was generated by Munin version 2.0.56 at 2023-05-22 23:15:11+0200 (CEST).

Section	Item	Metrics
Groups	localdomain	apache disk munin network postfix processes sensors system
	redhatnode	disk network processes system

En la sección *Groups* seleccionamos el nombre del nodo que estamos monitorizando, en nuestro caso *redhatnode*. De esta forma, se nos muestran una gran cantidad de métricas que podemos consultar de la máquina virtual Red Hat. A nosotros nos interesan concretamente las métricas que monitorizan el uso de la CPU, la memoria y operaciones de entrada-salida (IO). De los dos primeros tipos encontramos métricas en la sección *system*:

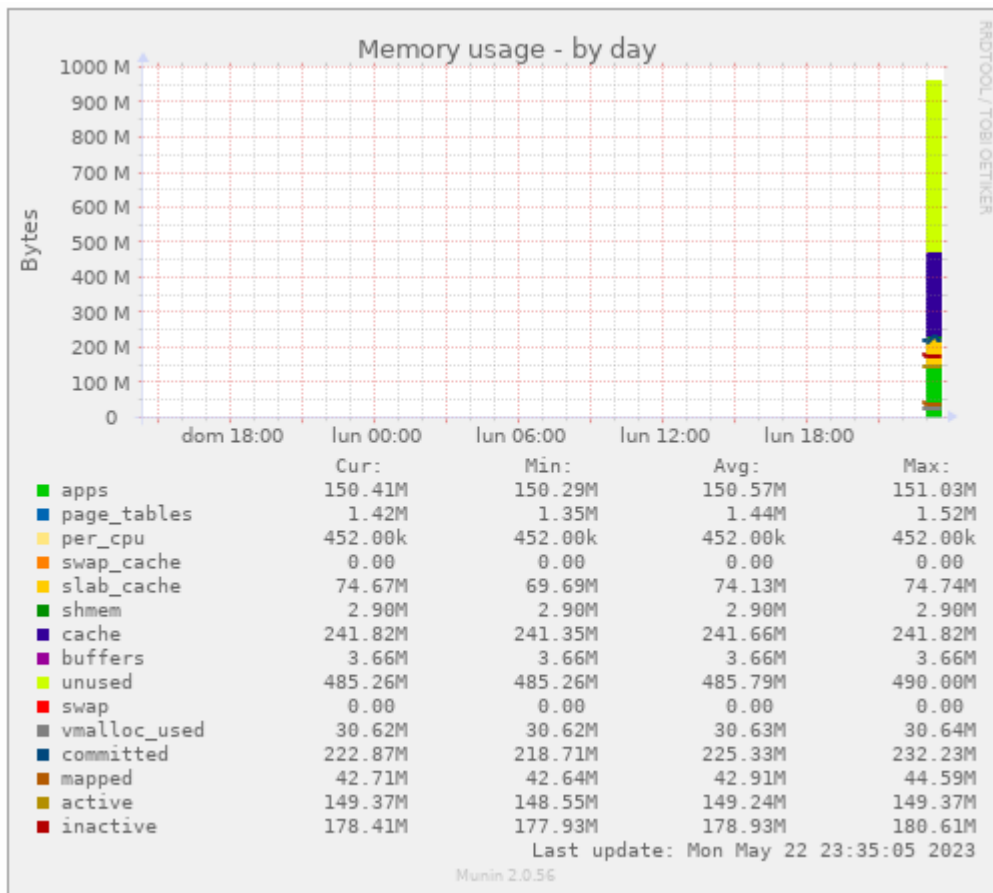
- system
  - Available entropy
  - CPU usage
  - File table usage
  - Individual interrupts
  - Inode table usage
  - Interrupts and context switches
  - Load average
  - Logged in users
  - Memory usage
  - SELinux Access Vector Cache
  - Swap in/out
  - Uptime

Comenzamos seleccionando *CPU usage* y se nos muestra la siguiente gráfica:



Como es lógico, si no estamos haciendo nada en la MV de Red Hat, en la gráfica se refleja que la CPU está prácticamente libre al 100% (*idle* significa que no está siendo usada por ningún programa). También se nos muestran gráficas por semana, mes y año, pero estas salen vacías ya que en mi caso la MV no lleva ni un día arrancada.

Seguimos con la monitorización de la memoria, para ello seleccionamos *Memory usage*:



En esta gráfica se nos muestra el reparto de la memoria. Como podemos ver, esta se reparte sobre todo entre memoria libre (unused), memoria caché y memoria compartida. También vemos que la memoria swap está a 0. También podemos comprobar que estos valores se corresponden con los de la MV si desde la terminal de esta misma ejecutamos *top* y *free*:

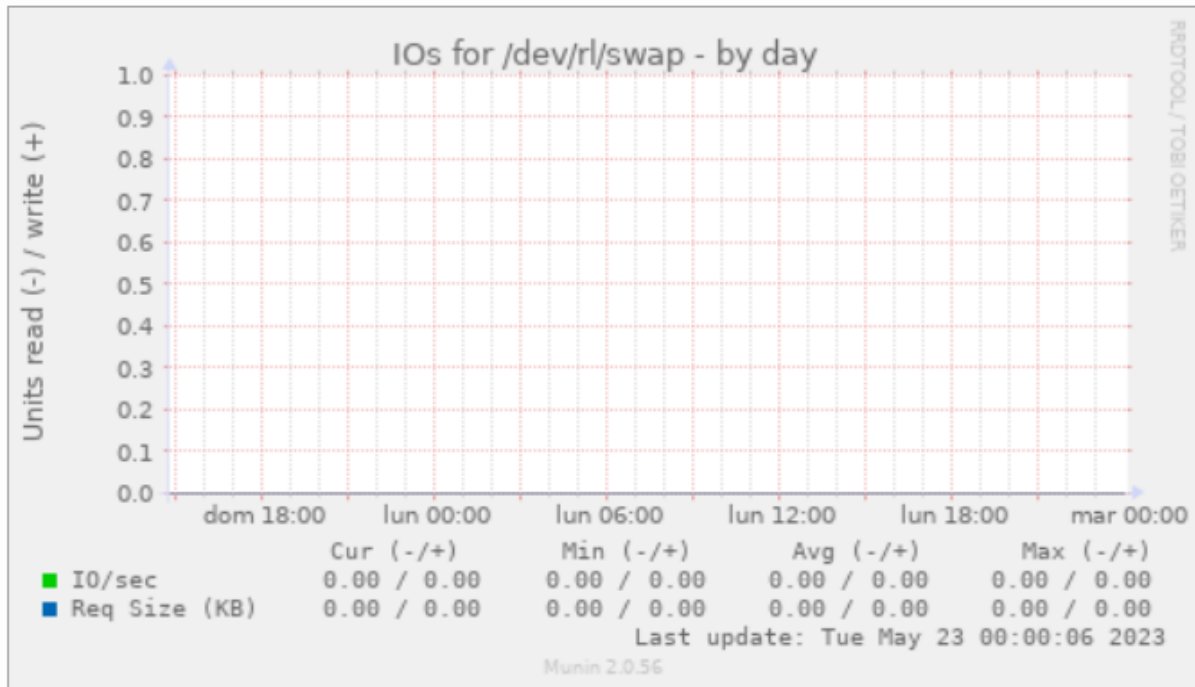
```
Tasks: 100 total, 1 running, 99 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.0 sy, 0.0 ni, 100.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 960.6 total, 485.3 free, 186.1 used, 289.2 buff/cache
MiB Swap: 820.0 total, 820.0 free, 0.0 used, 626.7 avail Mem
```

```
[jahoces@localhost:/etc/munin] 2023-05-22 23:46:56 $ free
              total        used        free      shared  buff/cache   available
Mem:      983644       190536       496908         2972       296200       641700
Swap:      839676           0       839676
```

Finalmente, vamos a usar una métrica IO. Nos fijamos en la sección *disk\_stats\_iops*:

- diskstats\_iops
  - disk
    - IOs for /dev/rl/root
    - IOs for /dev/rl/swap
    - IOs for /dev/sda

Para monitorizar las operaciones de entrada-salida en la partición `/dev/rl/swap` del disco, seleccionamos *IOs for /dev/rl/swap* y nos fijamos en la gráfica diaria:



Como era de esperar, no aparece nada en la gráfica ya que no se ha registrado ninguna operación de entrada o salida en el espacio de intercambio. Esto al fin y al cabo es lógico, pues ya habíamos visto que la memoria swap estaba libre al 100%.

## 4. Referencias

- <https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.raffaelechiatto.com%2Finstallare-e-configurare-munin-su-linux-debian%2F&psig=AOvVaw1kMzTobaIjXh2e3ZqV5oRx&ust=1686756701715000&source=images&cd=vfe&ved=0CBEQjRxqFwoTCOjrzsLIwP8CFQAAAAAdAAAAABAK> (Página de origen de la foto de la portada)
- <http://munin-monitoring.org/> (Página principal de Munin)
- <http://guide.munin-monitoring.org/en/latest/installation/install.html> (Página usada para la instalación)
- <http://guide.munin-monitoring.org/en/latest/installation/configuration.html> (Primera página usada para la configuración)
- <http://munin-monitoring.org/download/> (Segunda página usada para la configuración)