ACTIVIDAD 1 - EL TABLERO DE CONTROL

Cristóbal Suárez Abad

ADMINISTRACIÓN SISTEMAS GESTORES DE BASES DE DATOS - 2º ASIR

Índice

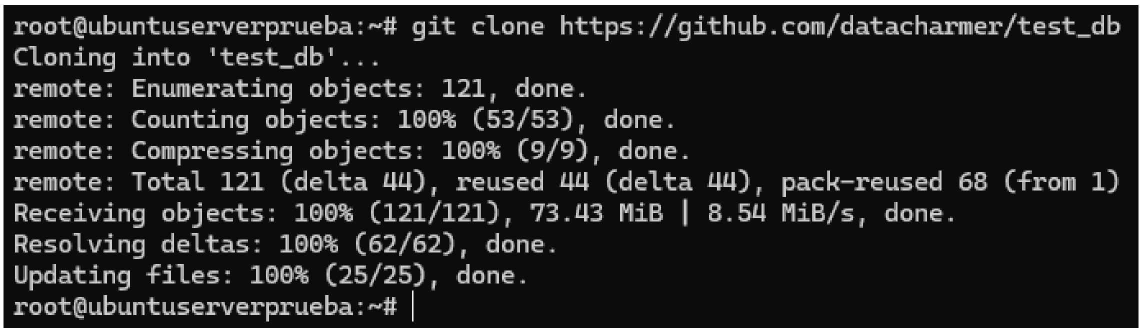
“Tenemos un servidor MySQL que va lento y queremos saber porque:”

1) En tu base de datos mysql, carga la base de datos de prueba Employees:

https://github.com/datacharmer/test_db

Usamos:

git clone https://github.com/datacharmer/test_db

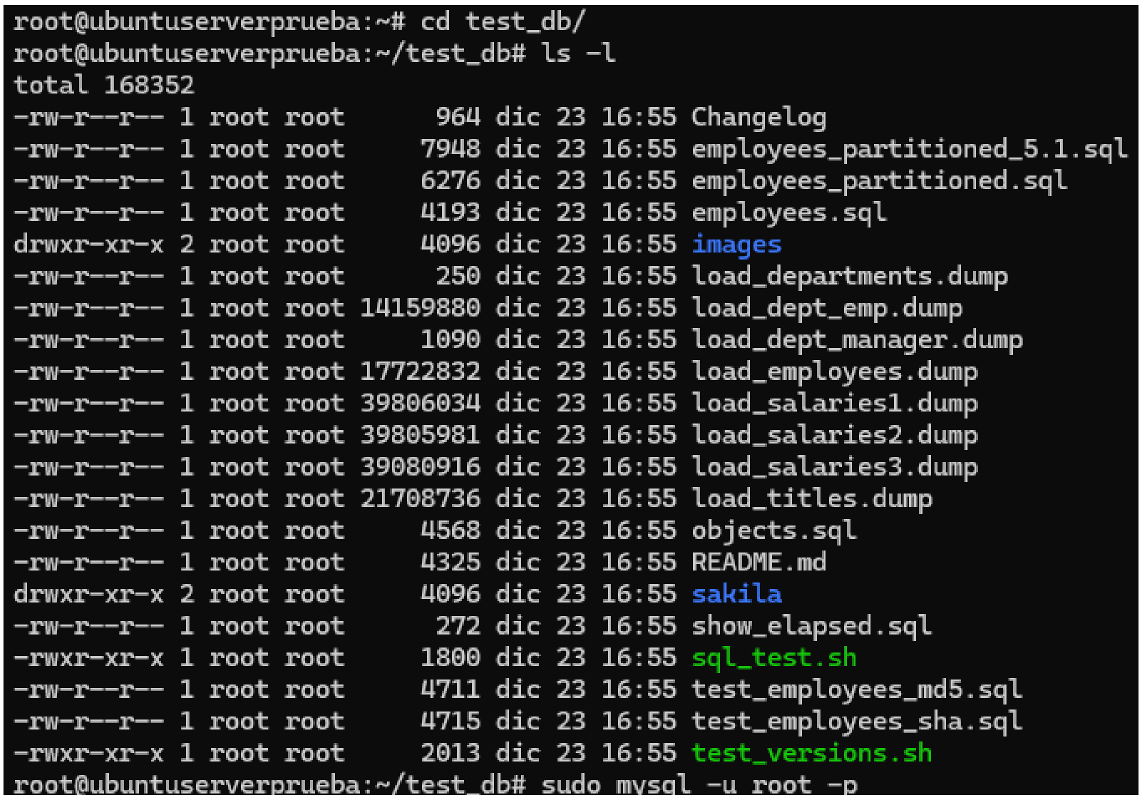


```
root@ubuntuserverprueba:~# git clone https://github.com/datacharmer/test_db
Cloning into 'test_db'...
remote: Enumerating objects: 121, done.
remote: Counting objects: 100% (53/53), done.
remote: Compressing objects: 100% (9/9), done.
remote: Total 121 (delta 44), reused 44 (delta 44), pack-reused 68 (from 1)
Receiving objects: 100% (121/121), 73.43 MiB | 8.54 MiB/s, done.
Resolving deltas: 100% (62/62), done.
Updating files: 100% (25/25), done.
root@ubuntuserverprueba:~# |
```

Entramos en mysql con un usuario con todos los privilegios.

sudo mysql -u root -p

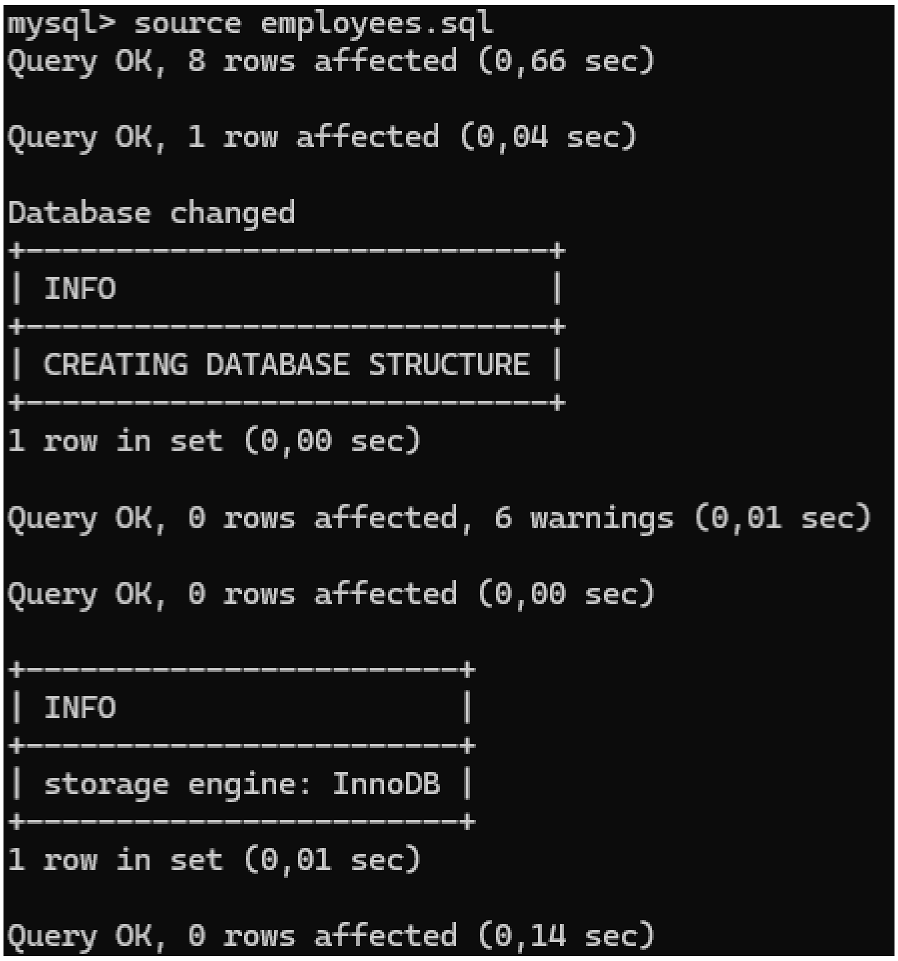
Y ejecutamos el script: Para ello se recomienda posicionarse en el directorio donde están los archivos antes de entrar en **mysql**.



```
root@ubuntuserverprueba:~# cd test_db/
root@ubuntuserverprueba:~/test_db# ls -l
total 168352
-rw-r--r-- 1 root root      964 dic 23 16:55 Changelog
-rw-r--r-- 1 root root     7948 dic 23 16:55 employees_partitioned_5.1.sql
-rw-r--r-- 1 root root     6276 dic 23 16:55 employees_partitioned.sql
-rw-r--r-- 1 root root     4193 dic 23 16:55 employees.sql
drwxr-xr-x 2 root root     4096 dic 23 16:55 images
-rw-r--r-- 1 root root      250 dic 23 16:55 load_departments.dump
-rw-r--r-- 1 root root 14159880 dic 23 16:55 load_dept_emp.dump
-rw-r--r-- 1 root root     1090 dic 23 16:55 load_dept_manager.dump
-rw-r--r-- 1 root root 17722832 dic 23 16:55 load_employees.dump
-rw-r--r-- 1 root root 39806034 dic 23 16:55 load_salaries1.dump
-rw-r--r-- 1 root root 39805981 dic 23 16:55 load_salaries2.dump
-rw-r--r-- 1 root root 39080916 dic 23 16:55 load_salaries3.dump
-rw-r--r-- 1 root root 21708736 dic 23 16:55 load_titles.dump
-rw-r--r-- 1 root root     4568 dic 23 16:55 objects.sql
-rw-r--r-- 1 root root     4325 dic 23 16:55 README.md
drwxr-xr-x 2 root root     4096 dic 23 16:55 sakila
-rw-r--r-- 1 root root      272 dic 23 16:55 show_elapsed.sql
-rwxr-xr-x 1 root root     1800 dic 23 16:55 sql_test.sh
-rw-r--r-- 1 root root     4711 dic 23 16:55 test_employees_md5.sql
-rw-r--r-- 1 root root     4715 dic 23 16:55 test_employees_sha.sql
-rwxr-xr-x 1 root root     2013 dic 23 16:55 test_versions.sh
root@ubuntuserverprueba:~/test_db# sudo mysql -u root -p
```

Luego usamos entramos en mysql y usamos source:

source employees.sql



```
mysql> source employees.sql
Query OK, 8 rows affected (0,66 sec)

Query OK, 1 row affected (0,04 sec)

Database changed
+-----------------------------+
| INFO                        |
+-----------------------------+
| CREATING DATABASE STRUCTURE |
+-----------------------------+
1 row in set (0,00 sec)

Query OK, 0 rows affected, 6 warnings (0,01 sec)

Query OK, 0 rows affected (0,00 sec)

+------------------------+
| INFO                   |
+------------------------+
| storage engine: InnoDB |
+------------------------+
1 row in set (0,01 sec)

Query OK, 0 rows affected (0,14 sec)
```

2) Crea un usuario para monitoreo:

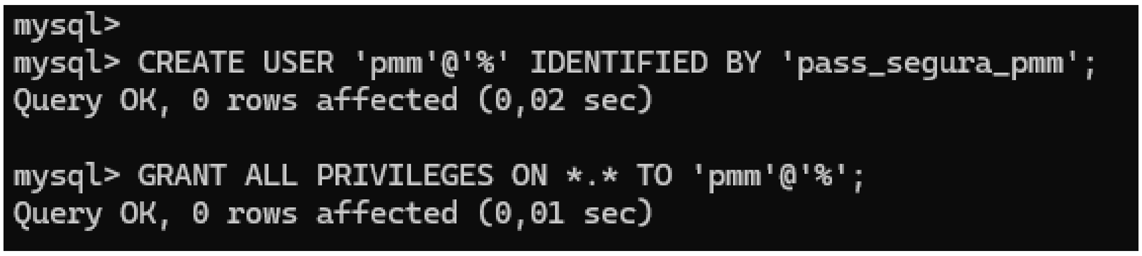
Entramos en “**mysql”**: **sudo mysql -u root -p**

CREATE USER 'pmm'@'%' IDENTIFIED BY 'pass_segura_pmm';

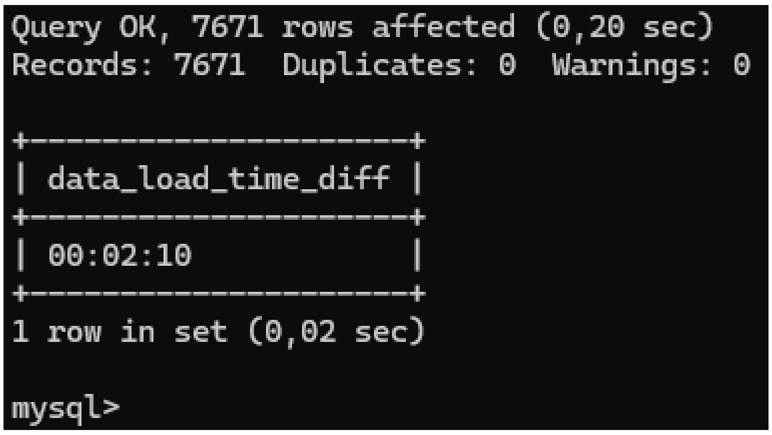
A Sergio le ha dado problemas con los permisos y al final ha recurrido a darle todos los privilegios posibles.

GRANT ALL PRIVILEGES ON *.* TO 'pmm'@'%';

FLUSH PRIVILEGES;



```
mysql>
mysql> CREATE USER 'pmm'@'%' IDENTIFIED BY 'pass_segura_pmm';
Query OK, 0 rows affected (0,02 sec)

mysql> GRANT ALL PRIVILEGES ON *.* TO 'pmm'@'%';
Query OK, 0 rows affected (0,01 sec)
```

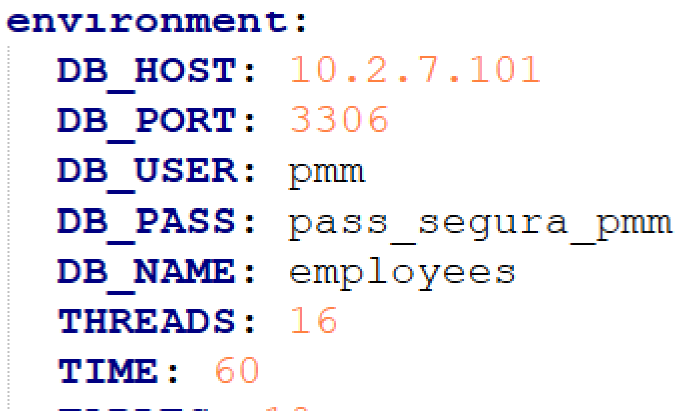


```
Query OK, 7671 rows affected (0,20 sec)
Records: 7671  Duplicates: 0  Warnings: 0

+----------------------+
| data_load_time_diff |
+----------------------+
| 00:02:10            |
+----------------------+
1 row in set (0,02 sec)

mysql>
```

3) Generación de Carga: Usar la herramienta Sysbench para estresar la base de datos (simular lecturas/escrituras masivas).

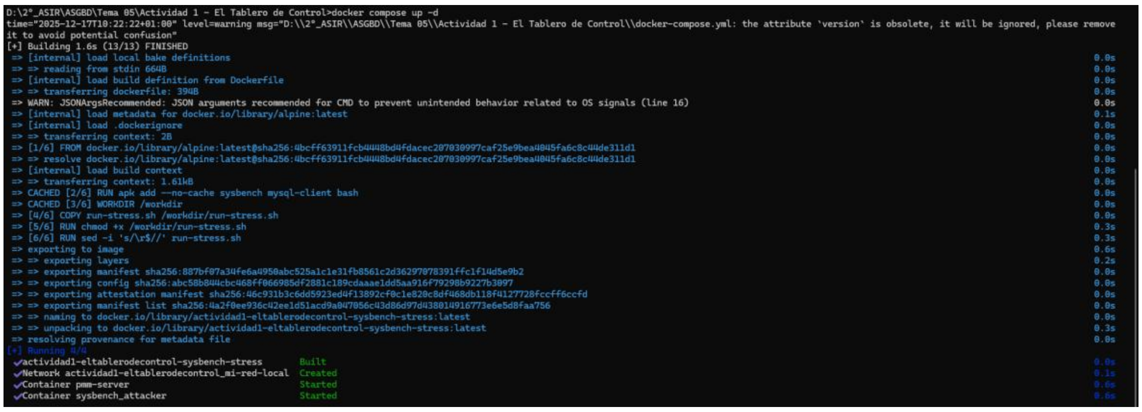
En los archivos adjuntos está desarrollado un contenedor para estresar tu base de datos (además de la aplicación PMM, para la parte 5).

a) Descarga el proyecto y cambia las opciones de conexión a la base de datos en docker-compose.yml



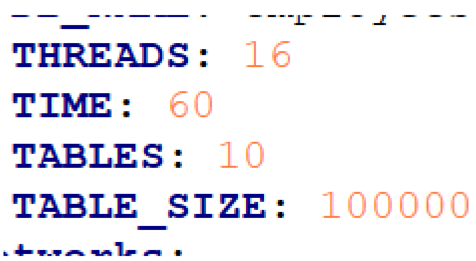
```
environment:
  DB_HOST: 10.2.7.101
  DB_PORT: 3306
  DB_USER: pmm
  DB_PASS: pass_segura_pmm
  DB_NAME: employees
  THREADS: 16
  TIME: 60
```

b) Ejecuta docker-compose up –build



```
D:\2°_ASIR\ASGBD\Tema 05\Actividad 1 - El Tablero de Control>docker compose up -d
time="2025-12-17T10:22:22+01:00" level=warning msg="D:\\2°_ASIR\\ASGBD\\Tema 05\\Actividad 1 - El Tablero de Control\\docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove
it to avoid potential confusion"
[+] Building 1.6s (13/13) FINISHED
 => [internal] load local bake definitions                                                                0.0s
 => => reading from stdin 664B                                                                            0.0s
 => [internal] load build definition from Dockerfile                                                      0.0s
 => => transferring dockerfile: 394B                                                                      0.0s
 => WARN: JSONArgsRecommended: JSON arguments recommended for CMD to prevent unintended behavior related to OS signals (line 16)   0.0s
 => [internal] load metadata for docker.io/library/alpine:latest                                          0.1s
 => [internal] load .dockerignore                                                                         0.0s
 => => transferring context: 2B                                                                           0.0s
 => [1/6] FROM docker.io/library/alpine:latest@sha256:4bcff63911fcb4448bd4fdacec207030997caf25e9bea4045fa6c8c44de311d1   0.0s
 => => resolve docker.io/library/alpine:latest@sha256:4bcff63911fcb4448bd4fdacec207030997caf25e9bea4045fa6c8c44de311d1  0.0s
 => [internal] load build context                                                                         0.0s
 => => transferring context: 1.61kB                                                                       0.0s
 => CACHED [2/6] RUN apk add --no-cache sysbench mysql-client bash                                        0.0s
 => CACHED [3/6] WORKDIR /workdir                                                                         0.0s
 => [4/6] COPY run-stress.sh /workdir/run-stress.sh                                                       0.0s
 => [5/6] RUN chmod +x /workdir/run-stress.sh                                                             0.3s
 => [6/6] RUN sed -i 's/\r$//' run-stress.sh                                                              0.3s
 => exporting to image                                                                                    0.6s
 => => exporting layers                                                                                   0.2s
 => => exporting manifest sha256:887bf07a34fe6a4950abc525a1c1e31fb8561c2d36297078391ffc1f14d5e9b2         0.0s
 => => exporting config sha256:abc58b844cbc468ff066985df2881c189cdaaae1dd5aa916f79298b9227b3097           0.0s
 => => exporting attestation manifest sha256:46c931b3c6dd5923ed4f13892cf0c1e820c8df468db118f4127728fccff6ccfd  0.0s
 => => exporting manifest list sha256:4a2f0ee936c42ee1d51acd9a047056c43d86d97d43801491677 3e6e5d8faa756   0.0s
 => => naming to docker.io/library/actividad1-eltablerodecontrol-sysbench-stress:latest                   0.0s
 => => unpacking to docker.io/library/actividad1-eltablerodecontrol-sysbench-stress:latest                0.3s
 => resolving provenance for metadata file                                                                0.0s
[+] Running 4/4
 ✔actividad1-eltablerodecontrol-sysbench-stress      Built                                                0.0s
 ✔Network actividad1-eltablerodecontrol_mi-red-local Created                                              0.1s
 ✔Container pmm-server                               Started                                              0.6s
 ✔Container sysbench_attacker                        Started                                              0.6s
```

Puedes modificar en docker-compose.yml la intensidad del ataque para adecuarlo a tu base de datos.



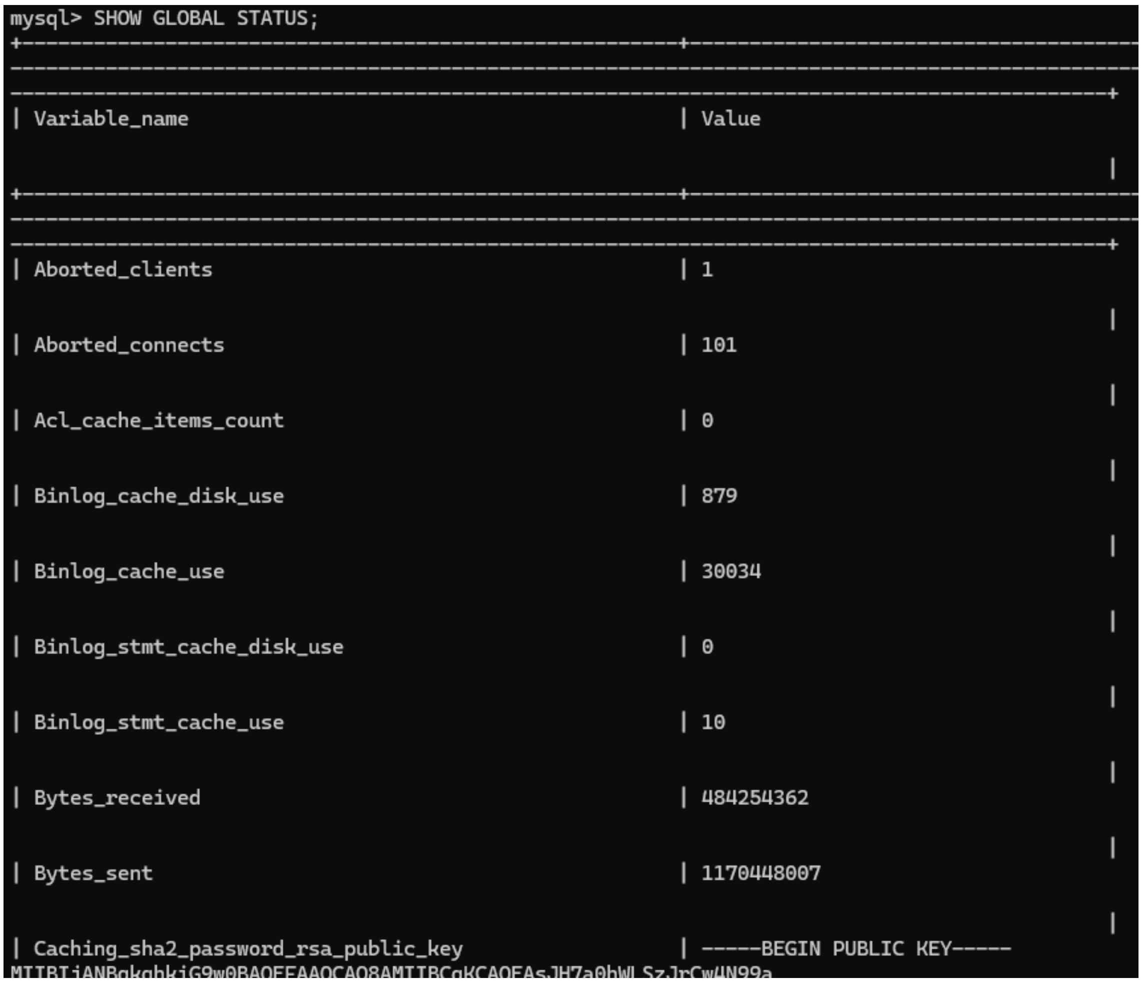
```
THREADS: 16
TIME: 60
TABLES: 10
TABLE_SIZE: 100000
```

4) Monitorización en vivo. Tener en cuenta que esté lanzado la generación de carga:

Usar comandos de consola: SHOW GLOBAL STATUS, mysqladmin -u root -p status -i 1, y top/htop en Linux. ¿Qué ves? ¿Para qué sirve cada uno de ellos?

- SHOW GLOBAL STATUS:

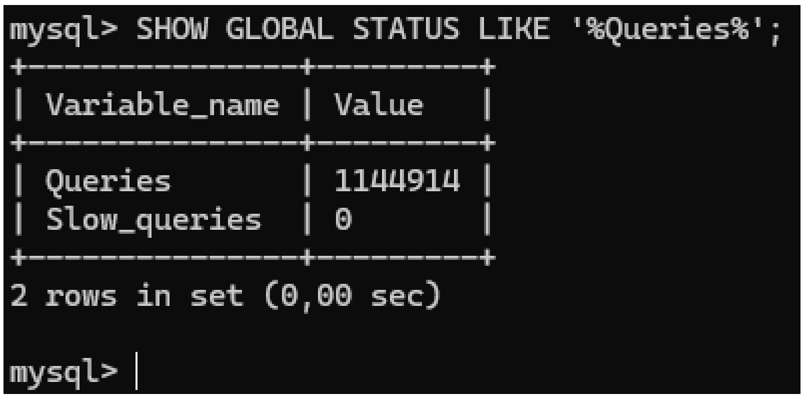
Desde dentro de **mysql**



```
mysql> SHOW GLOBAL STATUS;
+-------------------------------------------------------+--------------------------------------
----------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------+
| Variable_name                                          | Value
                                                                                           |
+-------------------------------------------------------+--------------------------------------
----------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------+
| Aborted_clients                                        | 1
                                                                                           |
| Aborted_connects                                       | 101
                                                                                           |
| Acl_cache_items_count                                  | 0
                                                                                           |
| Binlog_cache_disk_use                                  | 879
                                                                                           |
| Binlog_cache_use                                       | 30034
                                                                                           |
| Binlog_stmt_cache_disk_use                             | 0
                                                                                           |
| Binlog_stmt_cache_use                                  | 10
                                                                                           |
| Bytes_received                                         | 484254362
                                                                                           |
| Bytes_sent                                             | 1170448007
                                                                                           |
| Caching_sha2_password_rsa_public_key                   | -----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAsJH7a0bWLSzJrCw4N99a
```

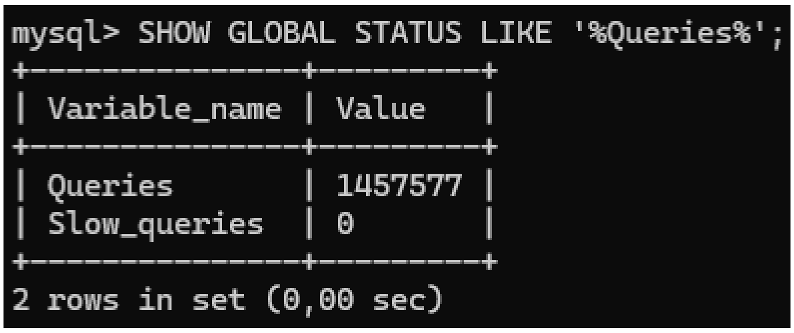
Nos da información sobre el estado de las operaciones del servidor. Es un poco más complejo de visionar lo que queremos, porque muestra mucha información. Podemos usar variantes de este comando como "SHOW GLOBAL STATUS LIKE '%Queries%';" que nos mostraría las "queries".

Antes del test:



```
mysql> SHOW GLOBAL STATUS LIKE '%Queries%';
+---------------+---------+
| Variable_name | Value   |
+---------------+---------+
| Queries       | 1144914 |
| Slow_queries  | 0       |
+---------------+---------+
2 rows in set (0,00 sec)

mysql> |
```

Después del test:



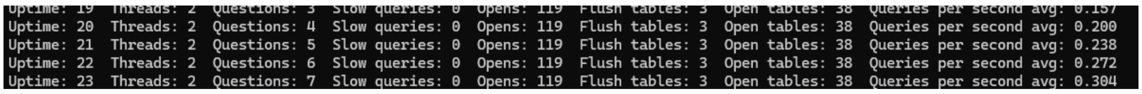
```
mysql> SHOW GLOBAL STATUS LIKE '%Queries%';
+---------------+---------+
| Variable_name | Value   |
+---------------+---------+
| Queries       | 1457577 |
| Slow_queries  | 0       |
+---------------+---------+
2 rows in set (0,00 sec)
```

El test ha realizado unas 312.663 queries.

- mysqladmin -u root -p status -i 1

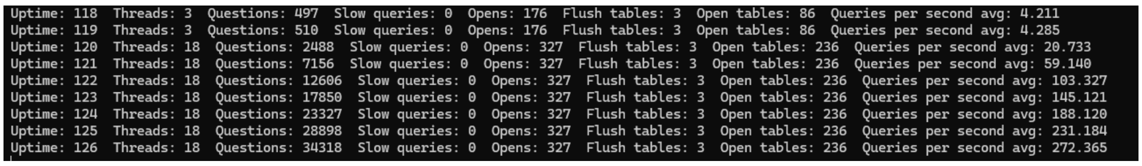
Muestra el tiempo de uso “uptime”, los hilos que se usan “threads”

Antes del test:



```
Uptime: 19  Threads: 2  Questions: 3  Slow queries: 0  Opens: 119  Flush tables: 3  Open tables: 38  Queries per second avg: 0.157
Uptime: 20  Threads: 2  Questions: 4  Slow queries: 0  Opens: 119  Flush tables: 3  Open tables: 38  Queries per second avg: 0.200
Uptime: 21  Threads: 2  Questions: 5  Slow queries: 0  Opens: 119  Flush tables: 3  Open tables: 38  Queries per second avg: 0.238
Uptime: 22  Threads: 2  Questions: 6  Slow queries: 0  Opens: 119  Flush tables: 3  Open tables: 38  Queries per second avg: 0.272
Uptime: 23  Threads: 2  Questions: 7  Slow queries: 0  Opens: 119  Flush tables: 3  Open tables: 38  Queries per second avg: 0.304
```

Después del test:

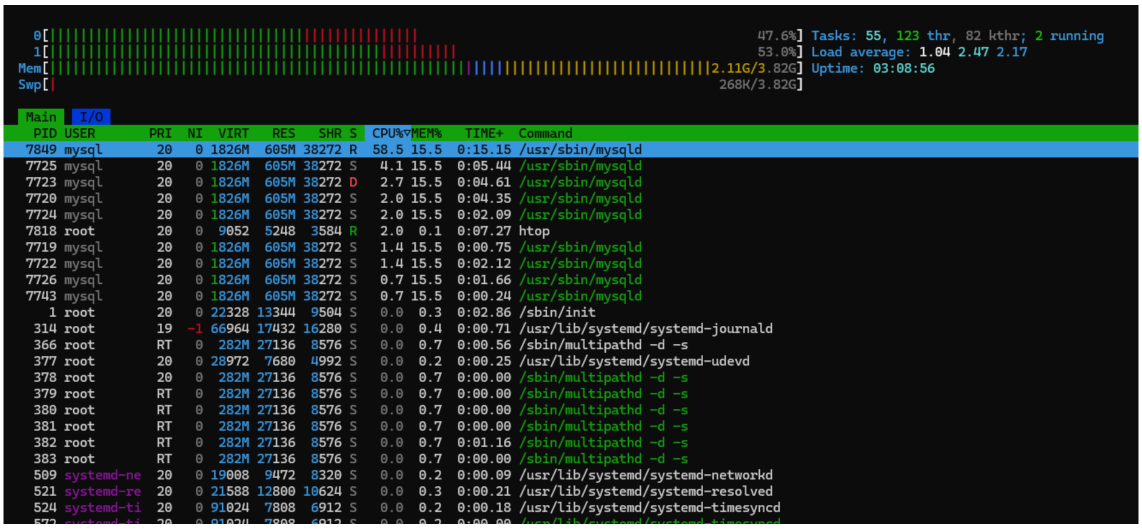


```
Uptime: 118  Threads: 3  Questions: 497  Slow queries: 0  Opens: 176  Flush tables: 3  Open tables: 86  Queries per second avg: 4.211
Uptime: 119  Threads: 3  Questions: 510  Slow queries: 0  Opens: 176  Flush tables: 3  Open tables: 86  Queries per second avg: 4.285
Uptime: 120  Threads: 18  Questions: 2488  Slow queries: 0  Opens: 327  Flush tables: 3  Open tables: 236  Queries per second avg: 20.733
Uptime: 121  Threads: 18  Questions: 7156  Slow queries: 0  Opens: 327  Flush tables: 3  Open tables: 236  Queries per second avg: 59.140
Uptime: 122  Threads: 18  Questions: 12606  Slow queries: 0  Opens: 327  Flush tables: 3  Open tables: 236  Queries per second avg: 103.327
Uptime: 123  Threads: 18  Questions: 17850  Slow queries: 0  Opens: 327  Flush tables: 3  Open tables: 236  Queries per second avg: 145.121
Uptime: 124  Threads: 18  Questions: 23327  Slow queries: 0  Opens: 327  Flush tables: 3  Open tables: 236  Queries per second avg: 188.120
Uptime: 125  Threads: 18  Questions: 28898  Slow queries: 0  Opens: 327  Flush tables: 3  Open tables: 236  Queries per second avg: 231.184
Uptime: 126  Threads: 18  Questions: 34318  Slow queries: 0  Opens: 327  Flush tables: 3  Open tables: 236  Queries per second avg: 272.365
```

La mayor diferencia es en el campo “**Queries per second avg**”, donde podemos ver el incremento de “**queries**”. El número de hilos, las **questions**, y las tablas abiertas.

- htop

Sirve para medir el consumo de CPU, RAM y otros componentes en el servidor. Indicando que procesos llevan a cabo ese consumo.



```
    0[||||||||||||||||||||||||||||||||||||||||||||||||||                              47.6%] Tasks: 55, 123 thr, 82 kthr; 2 running
    1[|||||||||||||||||||||||||||||||||||||||||||||||||||||||||                       53.0%] Load average: 1.04 2.47 2.17
  Mem[|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||2.11G/3.82G] Uptime: 03:08:56
  Swp[|                                                                          268K/3.82G]

  Main  I/O
    PID USER       PRI  NI  VIRT   RES   SHR S  CPU%▽MEM%   TIME+  Command
   7849 mysql       20   0 1826M  605M 38272 R  58.5 15.5  0:15.15 /usr/sbin/mysqld
   7725 mysql       20   0 1826M  605M 38272 S   4.1 15.5  0:05.44 /usr/sbin/mysqld
   7723 mysql       20   0 1826M  605M 38272 D   2.7 15.5  0:04.61 /usr/sbin/mysqld
   7720 mysql       20   0 1826M  605M 38272 S   2.0 15.5  0:04.35 /usr/sbin/mysqld
   7724 mysql       20   0 1826M  605M 38272 S   2.0 15.5  0:02.09 /usr/sbin/mysqld
   7818 root        20   0  9052  5248  3584 R   2.0  0.1  0:07.27 htop
   7719 mysql       20   0 1826M  605M 38272 S   1.4 15.5  0:00.75 /usr/sbin/mysqld
   7722 mysql       20   0 1826M  605M 38272 S   1.4 15.5  0:02.12 /usr/sbin/mysqld
   7726 mysql       20   0 1826M  605M 38272 S   0.7 15.5  0:01.66 /usr/sbin/mysqld
   7743 mysql       20   0 1826M  605M 38272 S   0.7 15.5  0:00.24 /usr/sbin/mysqld
      1 root        20   0 22328 13344  9504 S   0.0  0.3  0:02.86 /sbin/init
    314 root        19  -1 66964 17432 16280 S   0.0  0.4  0:00.71 /usr/lib/systemd/systemd-journald
    366 root        RT   0  282M 27136  8576 S   0.0  0.7  0:00.56 /sbin/multipathd -d -s
    377 root        20   0 28972  7680  4992 S   0.0  0.2  0:00.25 /usr/lib/systemd/systemd-udevd
    378 root        20   0  282M 27136  8576 S   0.0  0.7  0:00.00 /sbin/multipathd -d -s
    379 root        RT   0  282M 27136  8576 S   0.0  0.7  0:00.00 /sbin/multipathd -d -s
    380 root        RT   0  282M 27136  8576 S   0.0  0.7  0:00.00 /sbin/multipathd -d -s
    381 root        RT   0  282M 27136  8576 S   0.0  0.7  0:00.00 /sbin/multipathd -d -s
    382 root        RT   0  282M 27136  8576 S   0.0  0.7  0:01.16 /sbin/multipathd -d -s
    383 root        RT   0  282M 27136  8576 S   0.0  0.7  0:00.00 /sbin/multipathd -d -s
    509 systemd-ne  20   0 19008  9472  8320 S   0.0  0.2  0:00.09 /usr/lib/systemd/systemd-networkd
    521 systemd-re  20   0 21588 12800 10624 S   0.0  0.3  0:00.21 /usr/lib/systemd/systemd-resolved
    524 systemd-ti  20   0 91024  7808  6912 S   0.0  0.2  0:00.18 /usr/lib/systemd/systemd-timesyncd
```

Durante el testo, podemos observar que los procesos de mysql son los que más consumen.

Conectar MySQL Workbench y ver el dashboard de rendimiento (uso de CPU, tráfico de red, conexiones). Explica lo que ves.

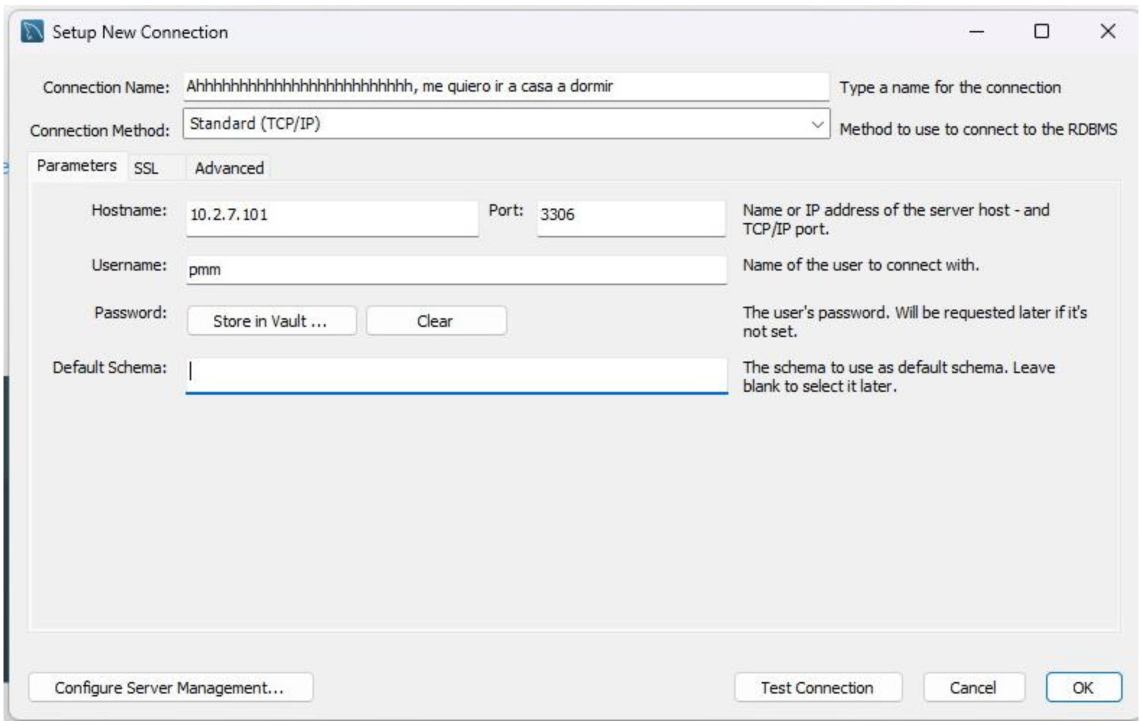
Nos descargamos el ejecutable desde la página web oficial:

https://dev.mysql.com/downloads/workbench/

E instalamos.

Abrimos MySQL Workbench y le damos a crear una nueva conexión.

Metemos los datos de la conexión: IP del servidor, credenciales del usuario y si queremos el nombre del esquema (no es necesario definirlo ahora)



Setup New Connection

Connection Name: Ahhhhhhhhhhhhhhhhhhhhhhhhh, me quiero ir a casa a dormir — Type a name for the connection

Connection Method: Standard (TCP/IP) — Method to use to connect to the RDBMS

Parameters | SSL | Advanced

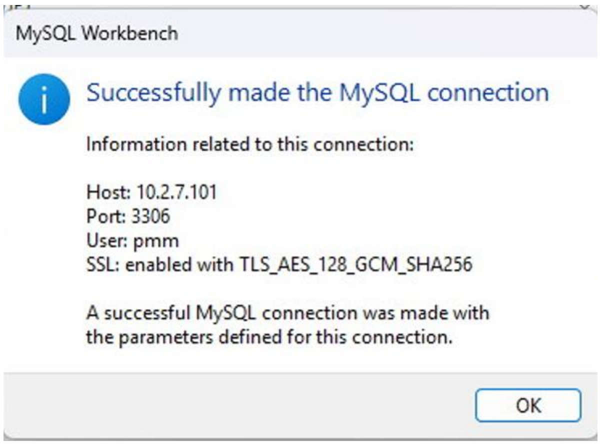
Hostname: 10.2.7.101 Port: 3306 — Name or IP address of the server host - and TCP/IP port.

Username: pmm — Name of the user to connect with.

Password: Store in Vault ... | Clear — The user's password. Will be requested later if it's not set.

Default Schema: — The schema to use as default schema. Leave blank to select it later.

Testeamos la conexión:


MySQL Workbench

Successfully made the MySQL connection

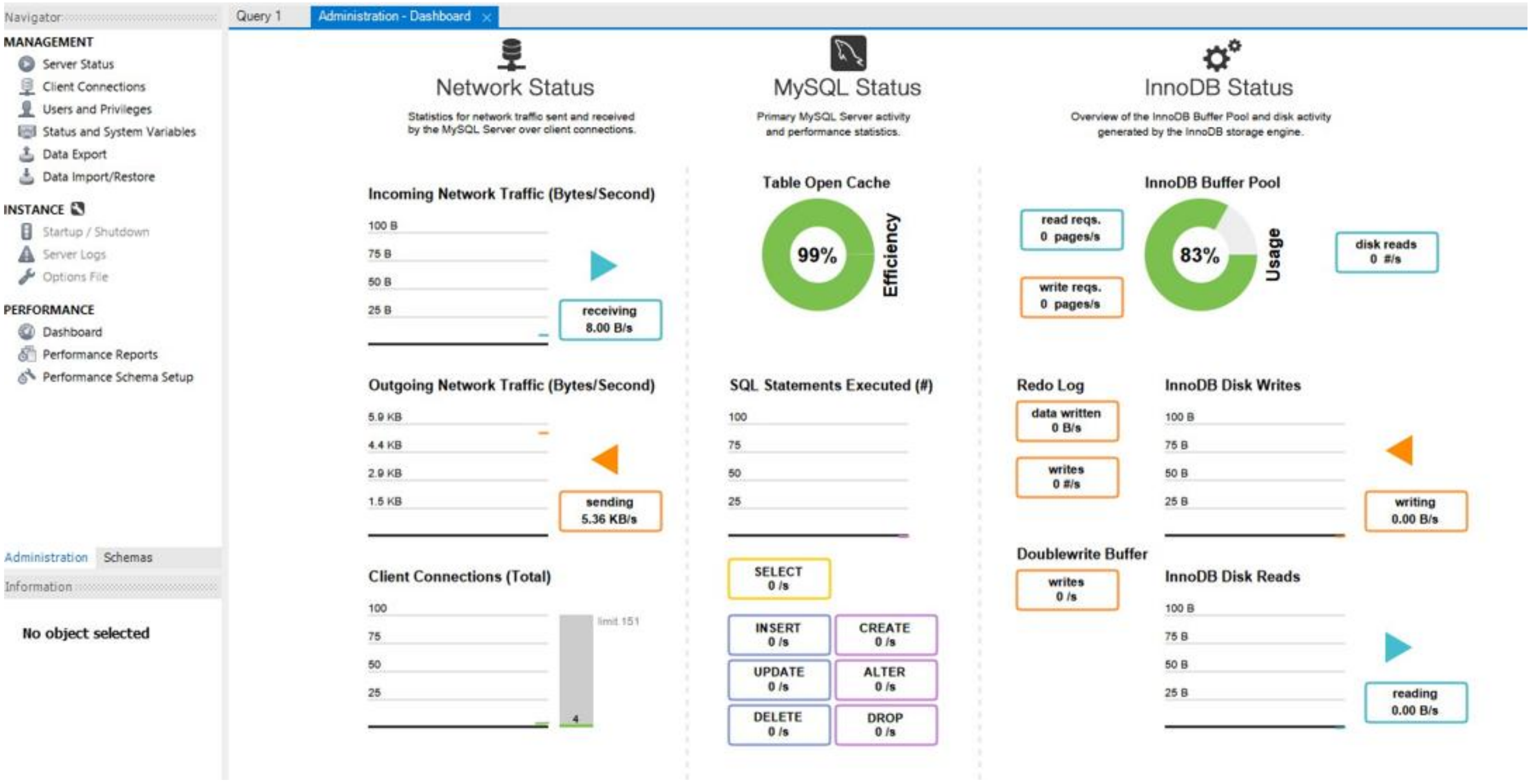
Information related to this connection:

Host: 10.2.7.101
Port: 3306
User: pmm
SSL: enabled with TLS_AES_128_GCM_SHA256

A successful MySQL connection was made with
the parameters defined for this connection.

Y guardamos.

Dashboard antes del test:



No object selected

Network Status

Statistics for network traffic sent and received by the MySQL Server over client connections.

MySQL Status

Primary MySQL Server activity and performance statistics.

InnoDB Status

Overview of the InnoDB Buffer Pool and disk activity generated by the InnoDB storage engine.

Incoming Network Traffic (Bytes/Second)

receiving 8.00 B/s

Outgoing Network Traffic (Bytes/Second)

sending 5.36 KB/s

Client Connections (Total)

Table Open Cache

99%

SQL Statements Executed (#)

SELECT 0 /s

INSERT 0 /s

CREATE 0 /s

UPDATE 0 /s

ALTER 0 /s

DELETE 0 /s

DROP 0 /s

InnoDB Buffer Pool

83%

read reqs. 0 pages/s

write reqs. 0 pages/s

disk reads 0 #/s

Redo Log

data written 0 B/s

writes 0 #/s

Doublewrite Buffer

writes 0 /s

InnoDB Disk Writes

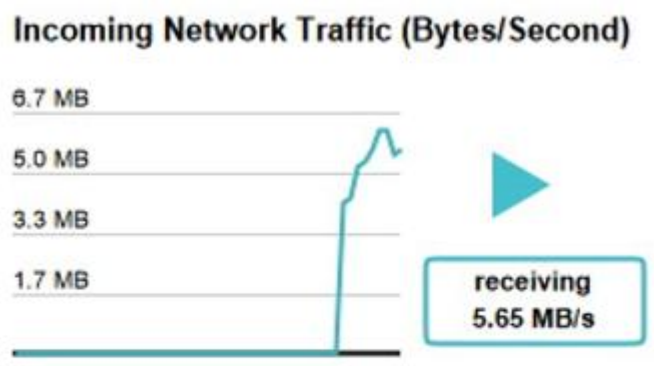
writing 0.00 B/s

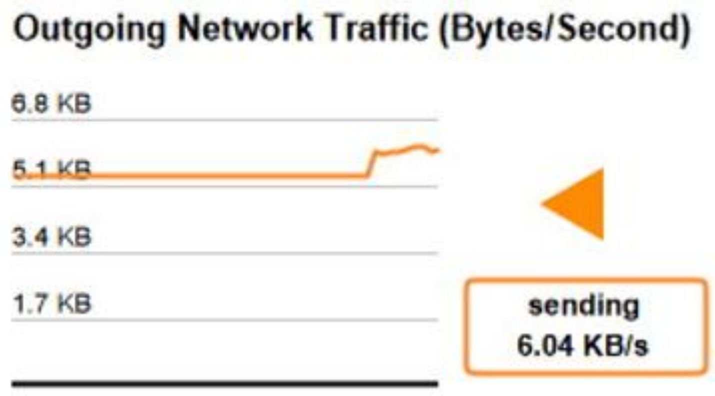
InnoDB Disk Reads

reading 0.00 B/s

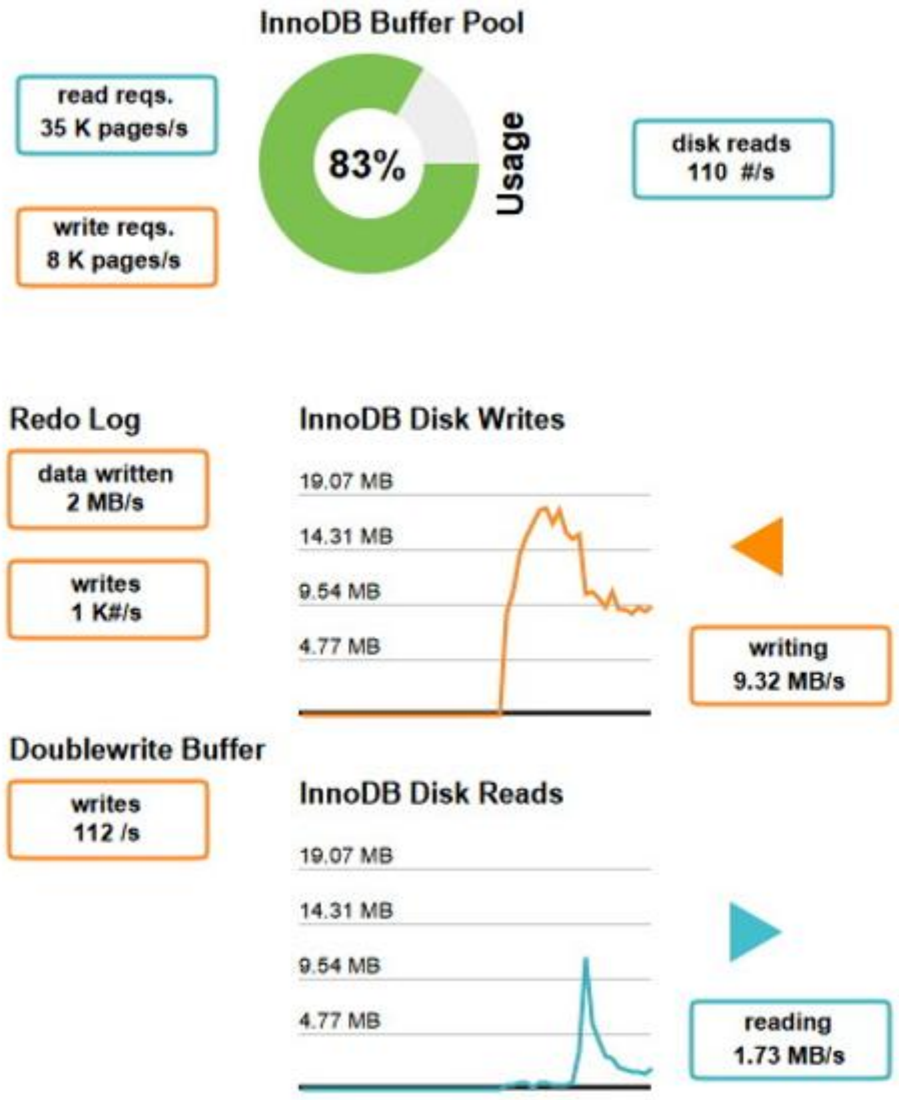
Durante el test:

- Incremento del tráfico de la Red. Más de entrada que de salida.

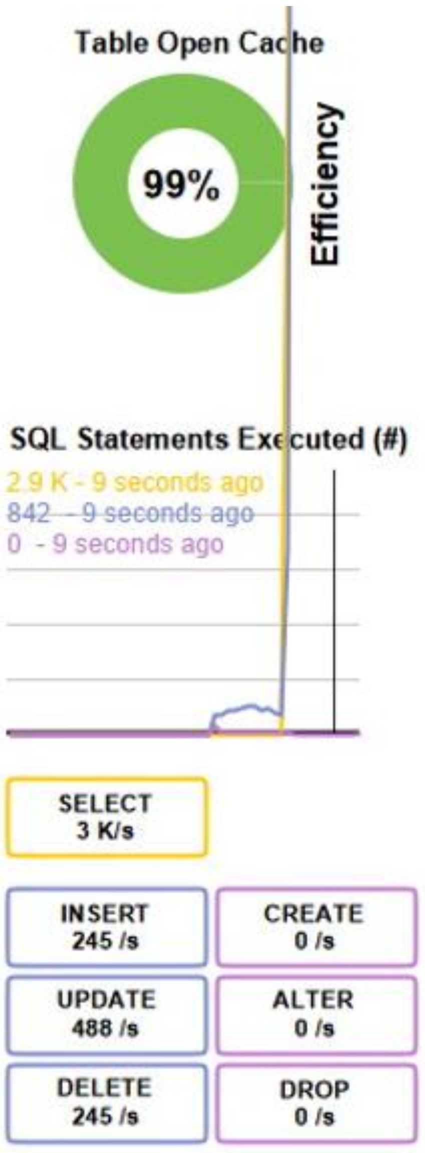




- Operaciones de lectura y escritura: Incremento inicial tanto de lectura como de escritura, manteniéndose el de escritura porque está introduciendo de nuevo los datos en las tablas.



- Statements: Se puede apreciar como hay un gran número de "deletes" al principio y luego se hacen "inserts" y "updates". El nivel es tan alto que se sale de la escala.



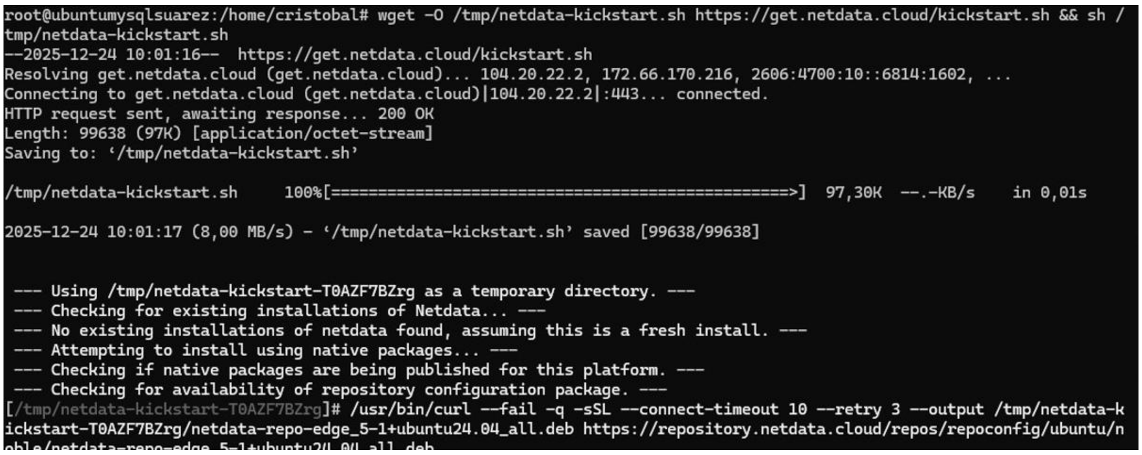
Instalar Netdata y ver las gráficas específicas de MySQL en tiempo real. Explica con detalle lo que ves.

Para instalar Netdata seguimos la guía:

https://www.tecmint.com/monitor-mysql-performance-with-netdata/

En el servidor donde tenemos la base de datos ponemos:

wget -O /tmp/netdata-kickstart.sh https://get.netdata.cloud/kickstart.sh && sh /tmp/netdata-kickstart.sh



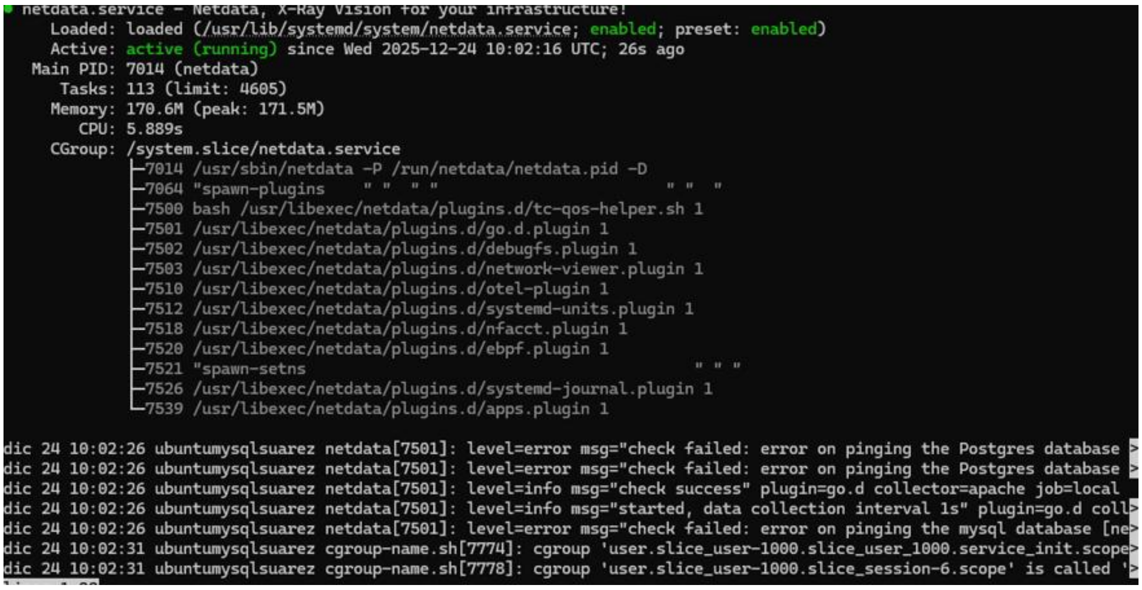
```
root@ubuntumysqlsuarez:/home/cristobal# wget -O /tmp/netdata-kickstart.sh https://get.netdata.cloud/kickstart.sh && sh /
tmp/netdata-kickstart.sh
--2025-12-24 10:01:16--  https://get.netdata.cloud/kickstart.sh
Resolving get.netdata.cloud (get.netdata.cloud)... 104.20.22.2, 172.66.170.216, 2606:4700:10::6814:1602, ...
Connecting to get.netdata.cloud (get.netdata.cloud)|104.20.22.2|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 99638 (97K) [application/octet-stream]
Saving to: '/tmp/netdata-kickstart.sh'

/tmp/netdata-kickstart.sh     100%[=================================================>]  97,30K  --.-KB/s    in 0,01s

2025-12-24 10:01:17 (8,00 MB/s) - '/tmp/netdata-kickstart.sh' saved [99638/99638]


 --- Using /tmp/netdata-kickstart-T0AZF7BZrg as a temporary directory. ---
 --- Checking for existing installations of Netdata... ---
 --- No existing installations of netdata found, assuming this is a fresh install. ---
 --- Attempting to install using native packages... ---
 --- Checking if native packages are being published for this platform. ---
 --- Checking for availability of repository configuration package. ---
[/tmp/netdata-kickstart-T0AZF7BZrg]# /usr/bin/curl --fail -q -sSL --connect-timeout 10 --retry 3 --output /tmp/netdata-k
ickstart-T0AZF7BZrg/netdata-repo-edge_5-1+ubuntu24.04_all.deb https://repository.netdata.cloud/repos/repoconfig/ubuntu/n
```

Lo iniciamos y habilitamos su arranque:

sudo systemctl start netdata

sudo systemctl status netdata

sudo systemctl enable netdata



```
● netdata.service - Netdata, X-Ray Vision for your infrastructure!
     Loaded: loaded (/usr/lib/systemd/system/netdata.service; enabled; preset: enabled)
     Active: active (running) since Wed 2025-12-24 10:02:16 UTC; 26s ago
   Main PID: 7014 (netdata)
      Tasks: 113 (limit: 4605)
     Memory: 170.6M (peak: 171.5M)
        CPU: 5.889s
     CGroup: /system.slice/netdata.service
             ├─7014 /usr/sbin/netdata -P /run/netdata/netdata.pid -D
             ├─7064 "spawn-plugins    " "  " "                                   " "  "
             ├─7500 bash /usr/libexec/netdata/plugins.d/tc-qos-helper.sh 1
             ├─7501 /usr/libexec/netdata/plugins.d/go.d.plugin 1
             ├─7502 /usr/libexec/netdata/plugins.d/debugfs.plugin 1
             ├─7503 /usr/libexec/netdata/plugins.d/network-viewer.plugin 1
             ├─7510 /usr/libexec/netdata/plugins.d/otel-plugin 1
             ├─7512 /usr/libexec/netdata/plugins.d/systemd-units.plugin 1
             ├─7518 /usr/libexec/netdata/plugins.d/nfacct.plugin 1
             ├─7520 /usr/libexec/netdata/plugins.d/ebpf.plugin 1
             ├─7521 "spawn-setns                                        " " " "
             ├─7526 /usr/libexec/netdata/plugins.d/systemd-journal.plugin 1
             └─7539 /usr/libexec/netdata/plugins.d/apps.plugin 1

dic 24 10:02:26 ubuntumysqlsuarez netdata[7501]: level=error msg="check failed: error on pinging the Postgres database >
dic 24 10:02:26 ubuntumysqlsuarez netdata[7501]: level=error msg="check failed: error on pinging the Postgres database >
dic 24 10:02:26 ubuntumysqlsuarez netdata[7501]: level=info msg="check success" plugin=go.d collector=apache job=local
dic 24 10:02:26 ubuntumysqlsuarez netdata[7501]: level=info msg="started, data collection interval 1s" plugin=go.d coll>
dic 24 10:02:26 ubuntumysqlsuarez netdata[7501]: level=error msg="check failed: error on pinging the mysql database [ne>
dic 24 10:02:31 ubuntumysqlsuarez cgroup-name.sh[7774]: cgroup 'user.slice_user-1000.slice_user_1000.service_init.scope>
dic 24 10:02:31 ubuntumysqlsuarez cgroup-name.sh[7778]: cgroup 'user.slice_user-1000.slice_session-6.scope' is called '>
```

En nuestro caso tenemos el **firewall** deshabilitado (seguridad ante todo, jejeje), pero para habilitar el puerto que usa Netdata tendríamos que usar:

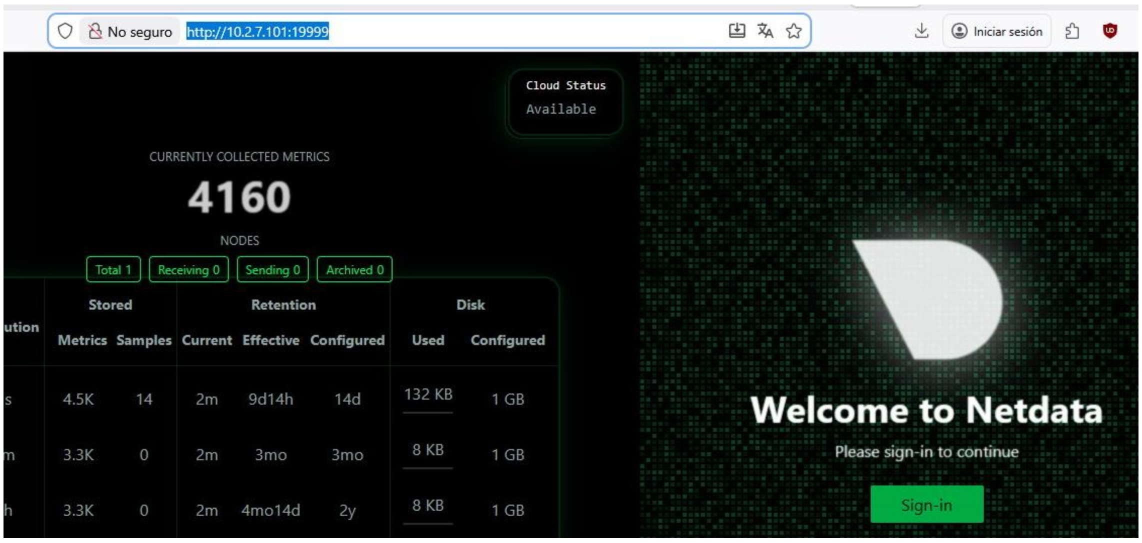
sudo firewall-cmd --permanent --add-port=19999/tcp

sudo firewall-cmd –reload

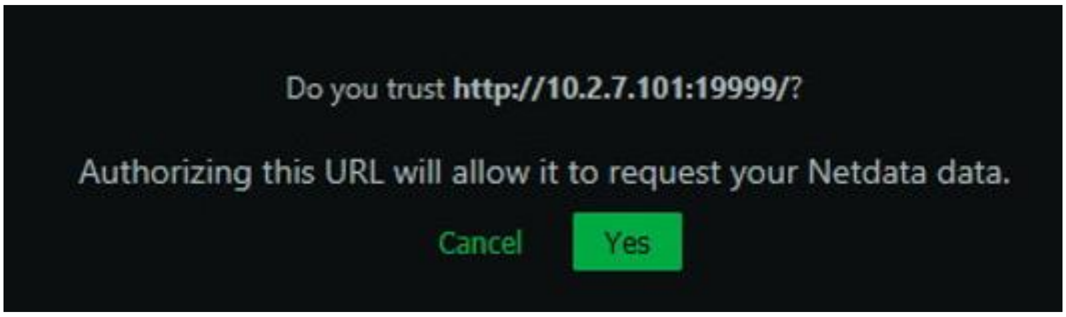
Una vez hecho eso podemos ir a nuestro navegador web y usar:

http://10.2.7.101:19999/

Debemos poner la ip del servidor y el puerto 19999.



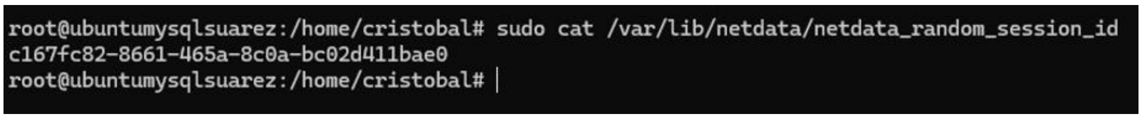
	Stored		Retention			Disk	
ution	Metrics	Samples	Current	Effective	Configured	Used	Configured
s	4.5K	14	2m	9d14h	14d	132 KB	1 GB
m	3.3K	0	2m	3mo	3mo	8 KB	1 GB
h	3.3K	0	2m	4mo14d	2y	8 KB	1 GB


Do you trust **http://10.2.7.101:19999/**?

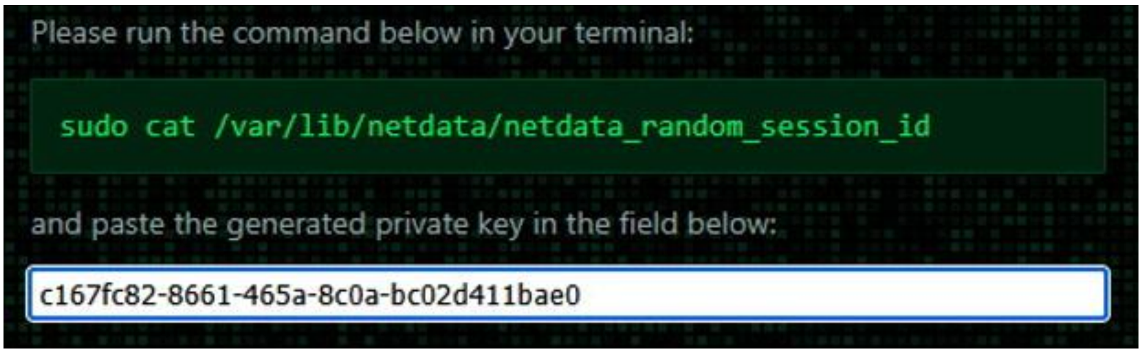
Authorizing this URL will allow it to request your Netdata data.

Nos pedirán que ejecutemos el siguiente comando en el servidor para generar un código:

sudo cat /var/lib/netdata/netdata_random_session_id



```
root@ubuntumysqlsuarez:/home/cristobal# sudo cat /var/lib/netdata/netdata_random_session_id
c167fc82-8661-465a-8c0a-bc02d411bae0
root@ubuntumysqlsuarez:/home/cristobal# |
```

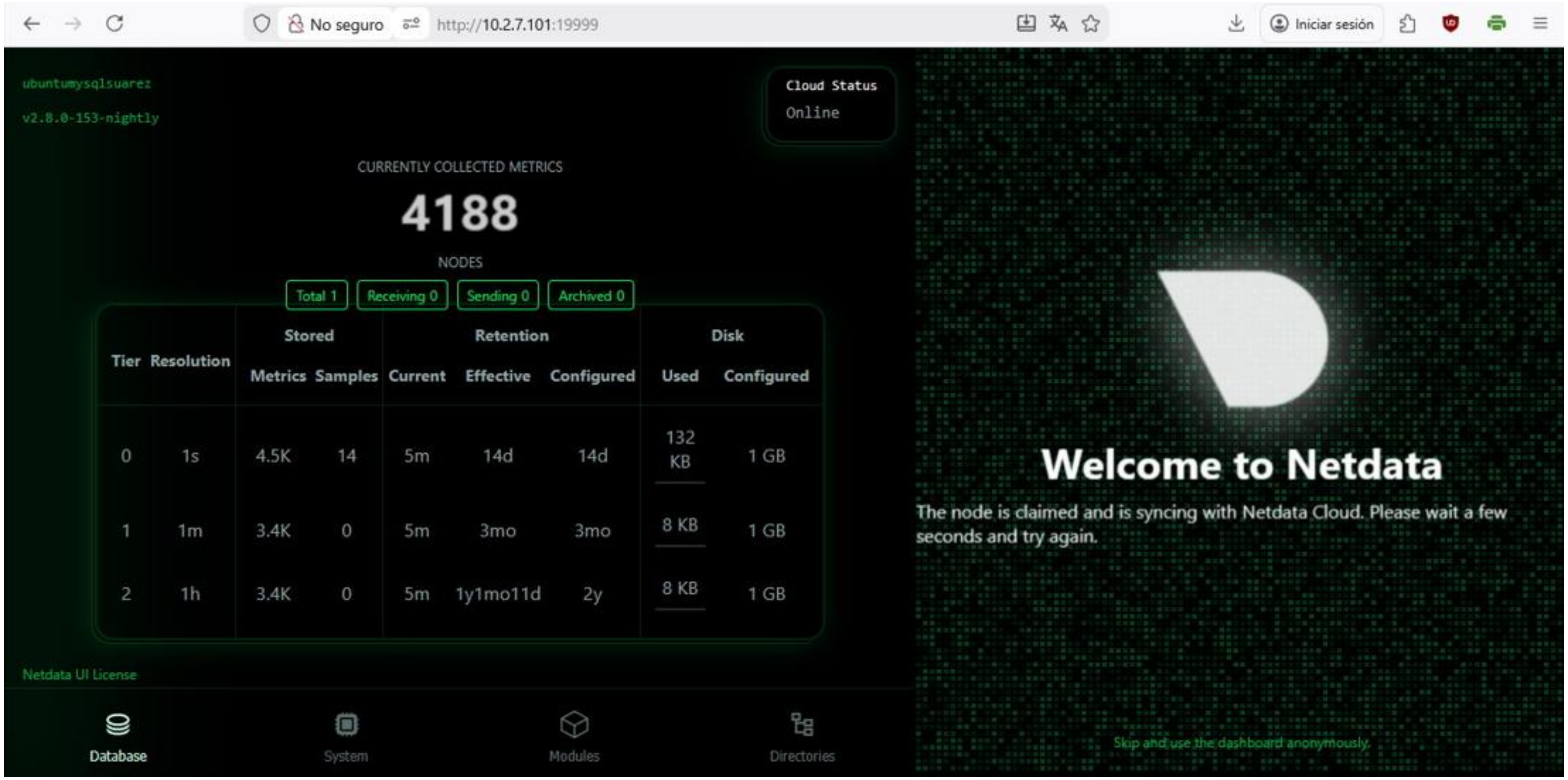


Please run the command below in your terminal:

```
sudo cat /var/lib/netdata/netdata_random_session_id
```

and paste the generated private key in the field below:

c167fc82-8661-465a-8c0a-bc02d411bae0



ubuntumysqlsuarez

v2.8.0-153-nightly

Cloud Status
Online

CURRENTLY COLLECTED METRICS

4188

NODES

Total 1 | Receiving 0 | Sending 0 | Archived 0

Tier	Resolution	Stored		Retention			Disk	
		Metrics	Samples	Current	Effective	Configured	Used	Configured
0	1s	4.5K	14	5m	14d	14d	132 KB	1 GB
1	1m	3.4K	0	5m	3mo	3mo	8 KB	1 GB
2	1h	3.4K	0	5m	1y1mo11d	2y	8 KB	1 GB

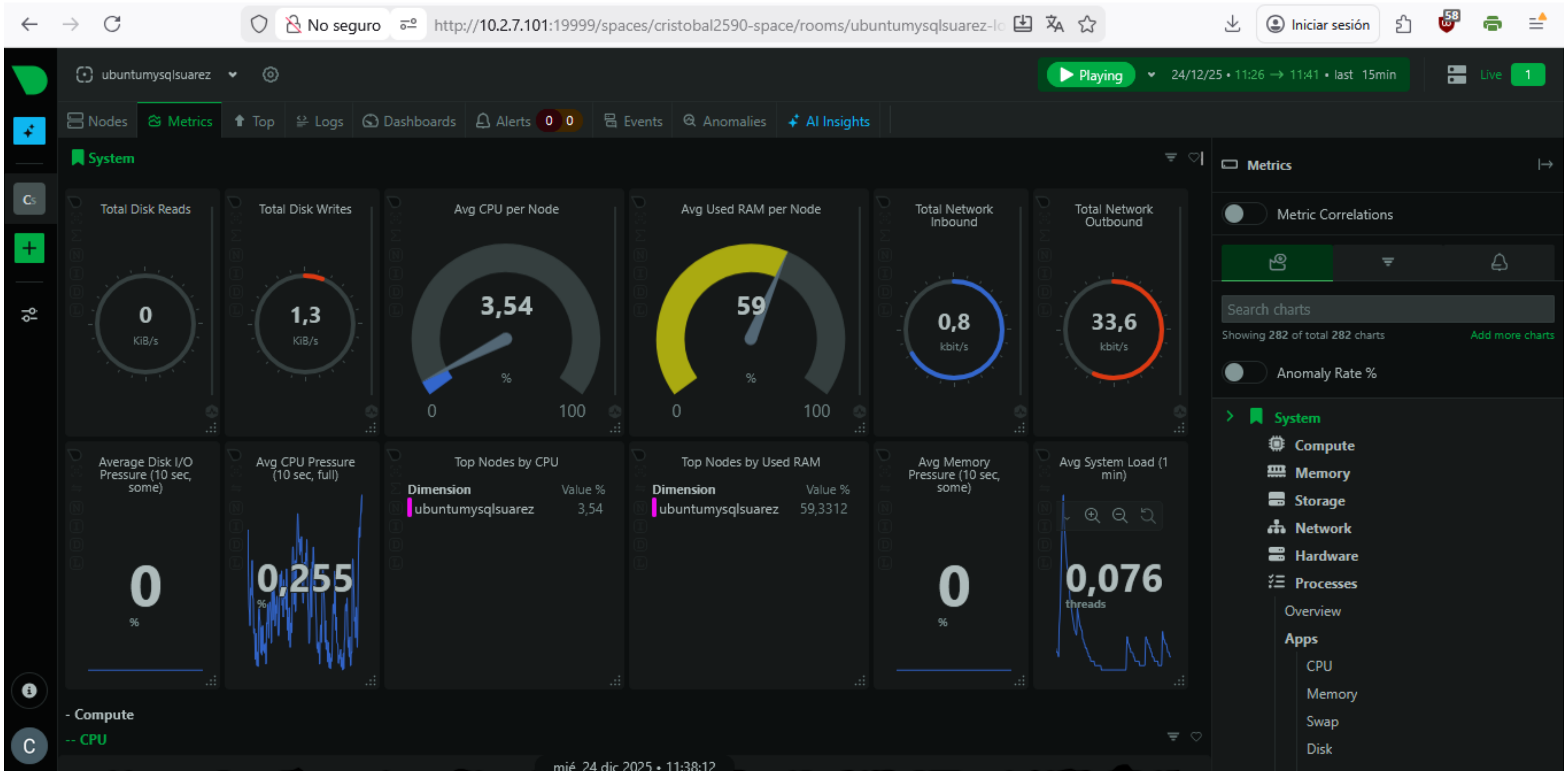
Netdata UI License

Welcome to Netdata

The node is claimed and is syncing with Netdata Cloud. Please wait a few seconds and try again.

Skip and use the dashboard anonymously.

Plan Premium de 14 días gratuitos. En el momento de la captura el servidor está en reposo.


No seguro
http://10.2.7.101:19999/spaces/cristobal2590-space/rooms/ubuntumysqlsuarez-lo
Iniciar sesión
ubuntumysqlsuarez
Playing
24/12/25 • 11:26 → 11:41 • last 15min
Live 1
Nodes
Metrics
Top
Logs
Dashboards
Alerts 0 0
Events
Anomalies
AI Insights
System
Total Disk Reads 0 KiB/s
Total Disk Writes 1,3 KiB/s
Avg CPU per Node 3,54 % 0 100
Avg Used RAM per Node 59 % 0 100
Total Network Inbound 0,8 kbit/s
Total Network Outbound 33,6 kbit/s
Average Disk I/O Pressure (10 sec, some) 0 %
Avg CPU Pressure (10 sec, full) 0,255 %
Top Nodes by CPU
Dimension Value %
ubuntumysqlsuarez 3,54
Top Nodes by Used RAM
Dimension Value %
ubuntumysqlsuarez 59,3312
Avg Memory Pressure (10 sec, some) 0 %
Avg System Load (1 min) 0,076 threads
- Compute
-- CPU
mié. 24 dic 2025 • 11:38:12
Metrics
Metric Correlations
Search charts
Showing 282 of total 282 charts
Add more charts
Anomaly Rate %
System
Compute
Memory
Storage
Network
Hardware
Processes
Overview
Apps
CPU
Memory
Swap
Disk

Ahora con más movimiento.

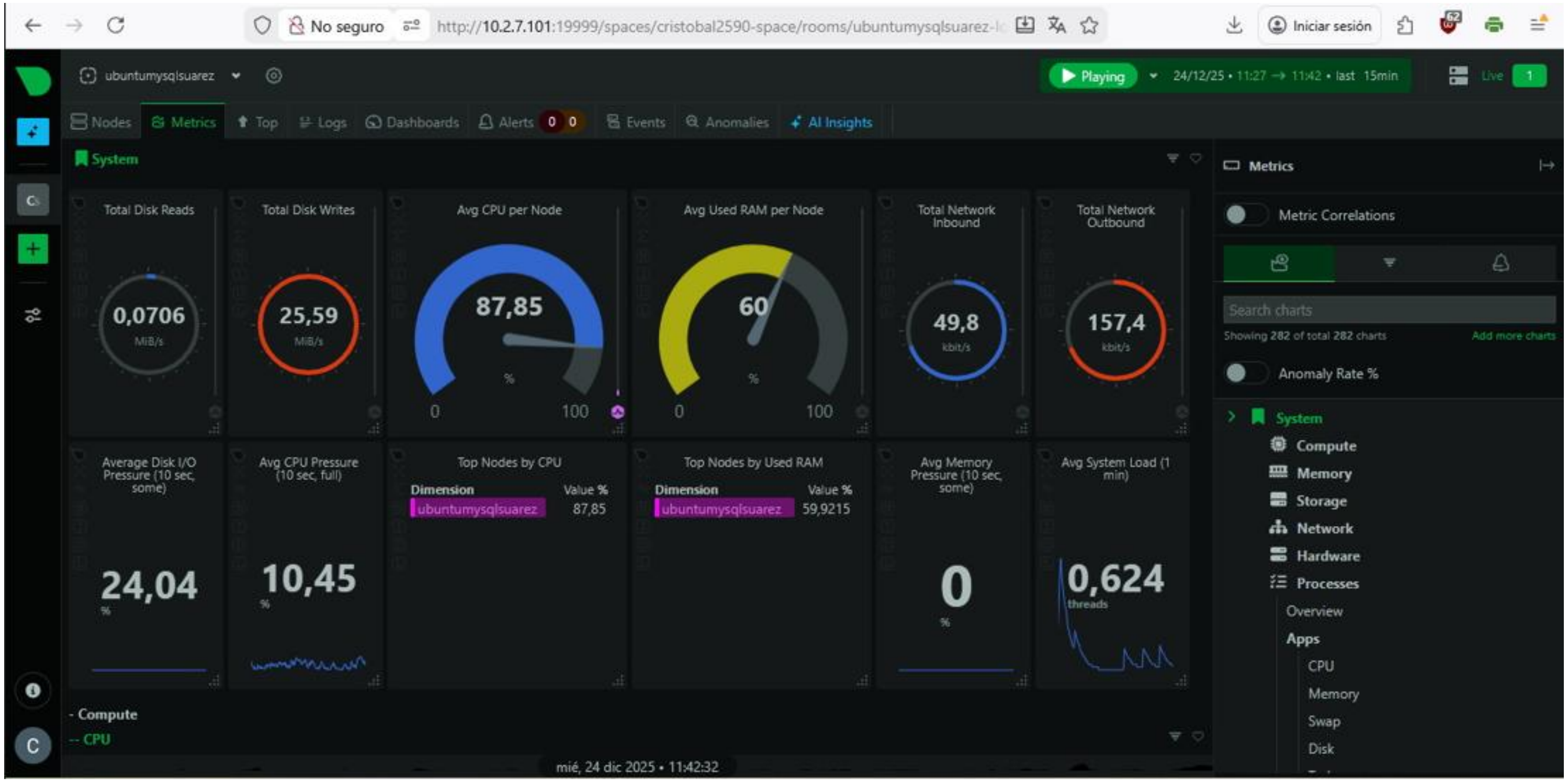


Chart	Value
Total Disk Reads	0,0706 MiB/s
Total Disk Writes	25,59 MiB/s
Avg CPU per Node	87,85 %
Avg Used RAM per Node	60 %
Total Network Inbound	49,8 kbit/s
Total Network Outbound	157,4 kbit/s
Average Disk I/O Pressure (10 sec, some)	24,04 %
Avg CPU Pressure (10 sec, full)	10,45 %
Avg Memory Pressure (10 sec, some)	0 %
Avg System Load (1 min)	0,624 threads

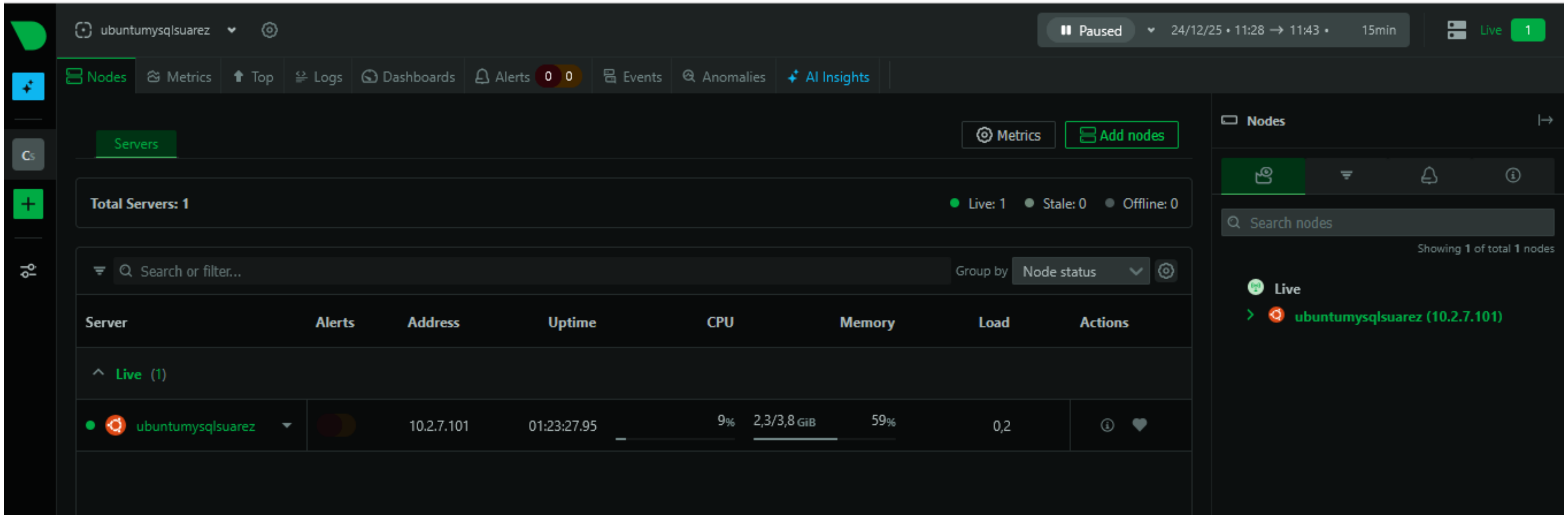
Top Nodes by CPU

Dimension	Value %
ubuntumysqlsuarez	87,85

Top Nodes by Used RAM

Dimension	Value %
ubuntumysqlsuarez	59,9215

Nos muestra los nodos que tenemos vinculados a Netdata:

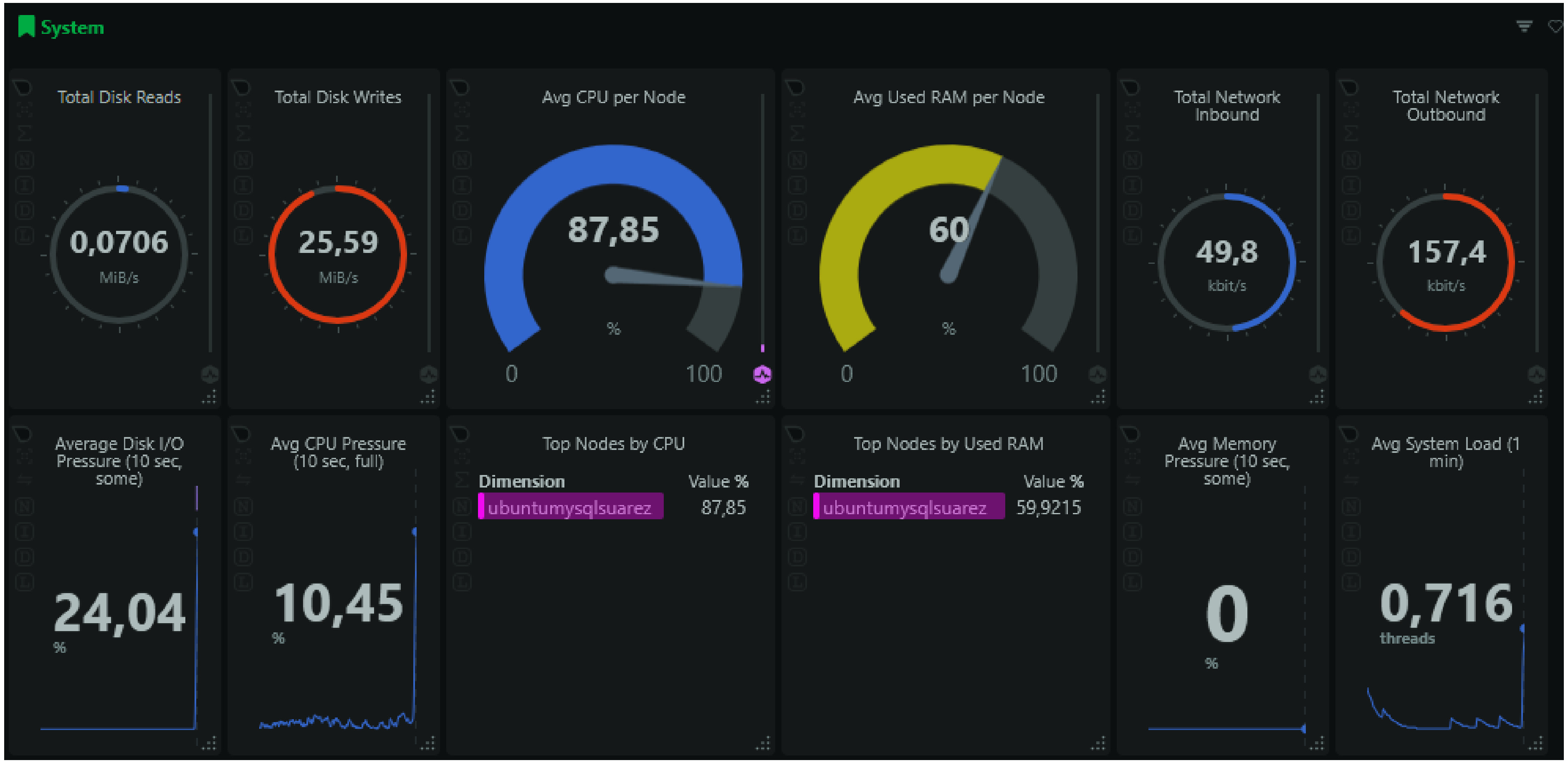


Total Servers: 1 ● Live: 1 ● Stale: 0 ● Offline: 0

Search or filter... Group by Node status

Server	Alerts	Address	Uptime	CPU	Memory	Load	Actions
Live (1)							
ubuntumysqlsuarez		10.2.7.101	01:23:27.95	9%	2,3/3,8 GiB 59%	0,2	

Resumen de métricas del nodo elegido:



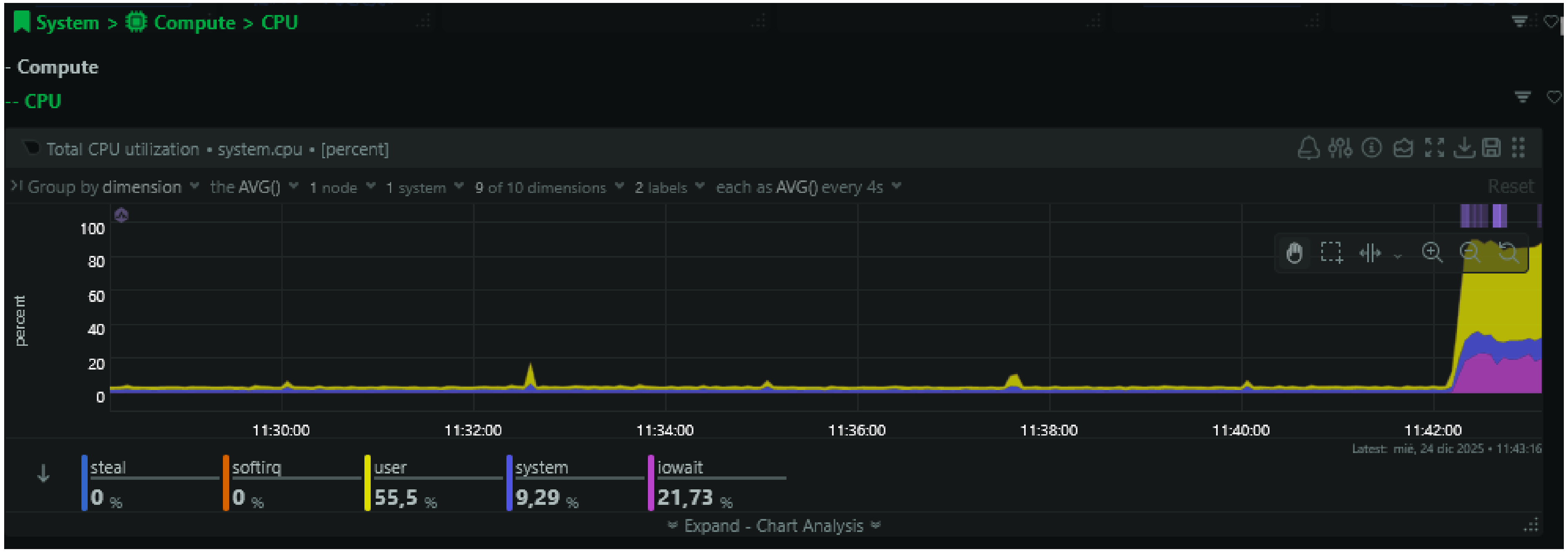
Top Nodes by CPU

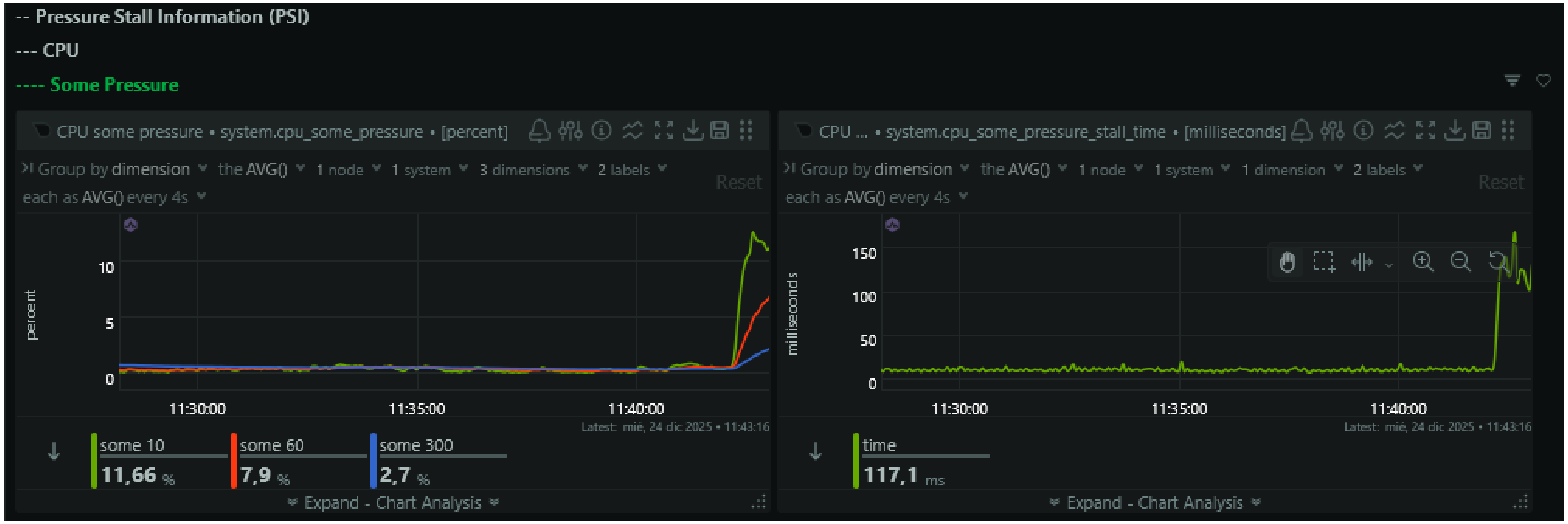
Dimension	Value %
ubuntumysqlsuarez	87,85

Top Nodes by Used RAM

Dimension	Value %
ubuntumysqlsuarez	59,9215

Diagrama con el resumen del uso de la CPU:





-- Pressure Stall Information (PSI)

--- CPU

---- Some Pressure

CPU some pressure • system.cpu_some_pressure • [percent]

Group by dimension the AVG() 1 node 1 system 3 dimensions 2 labels Reset

each as AVG() every 4s

Latest: mié, 24 dic 2025 • 11:43:16

some 10 11,66 %
some 60 7,9 %
some 300 2,7 %

CPU ... • system.cpu_some_pressure_stall_time • [milliseconds]

Group by dimension the AVG() 1 node 1 system 1 dimension 2 labels Reset

each as AVG() every 4s

Latest: mié, 24 dic 2025 • 11:43:16

time 117,1 ms

De la RAM:

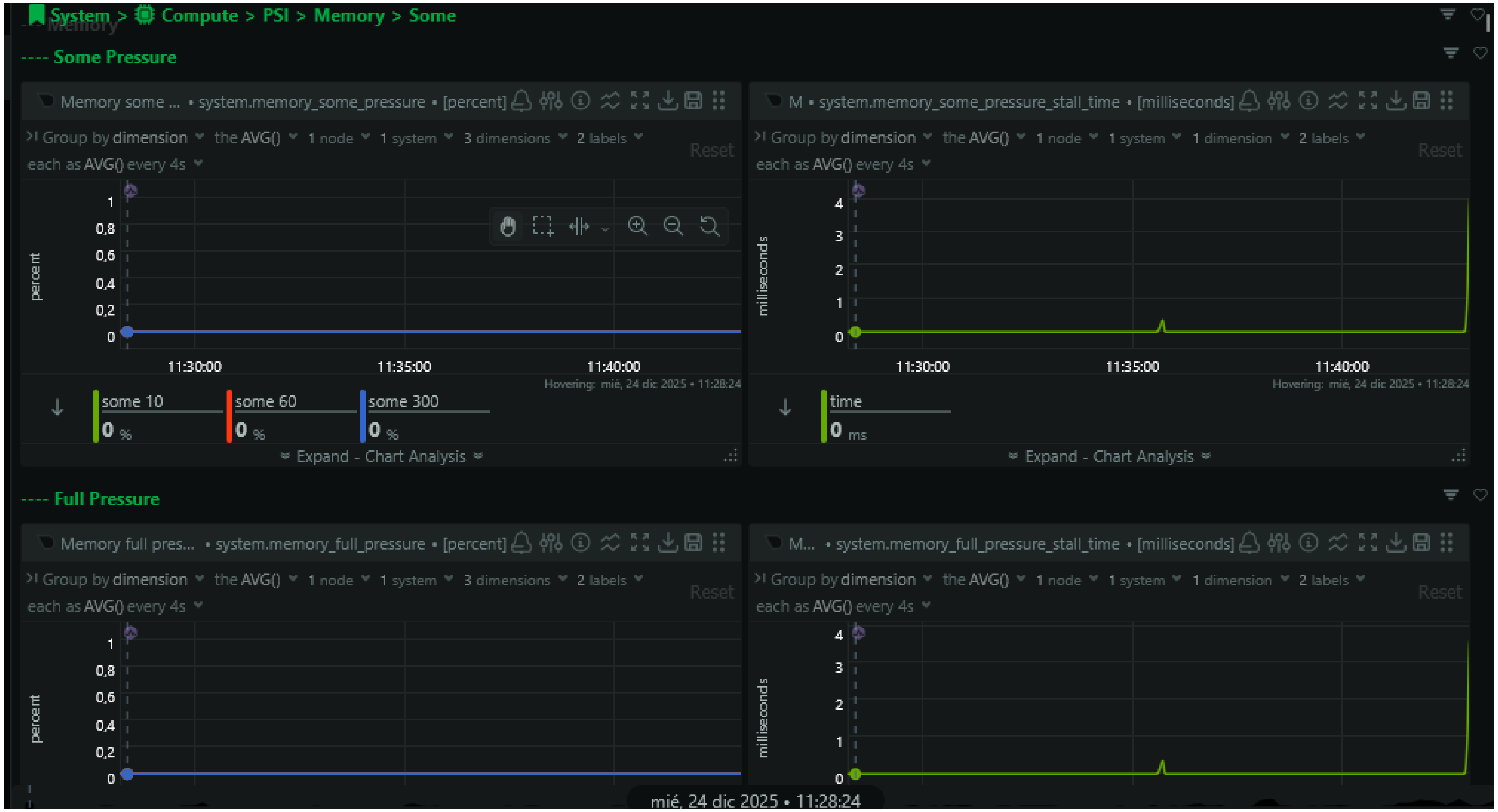
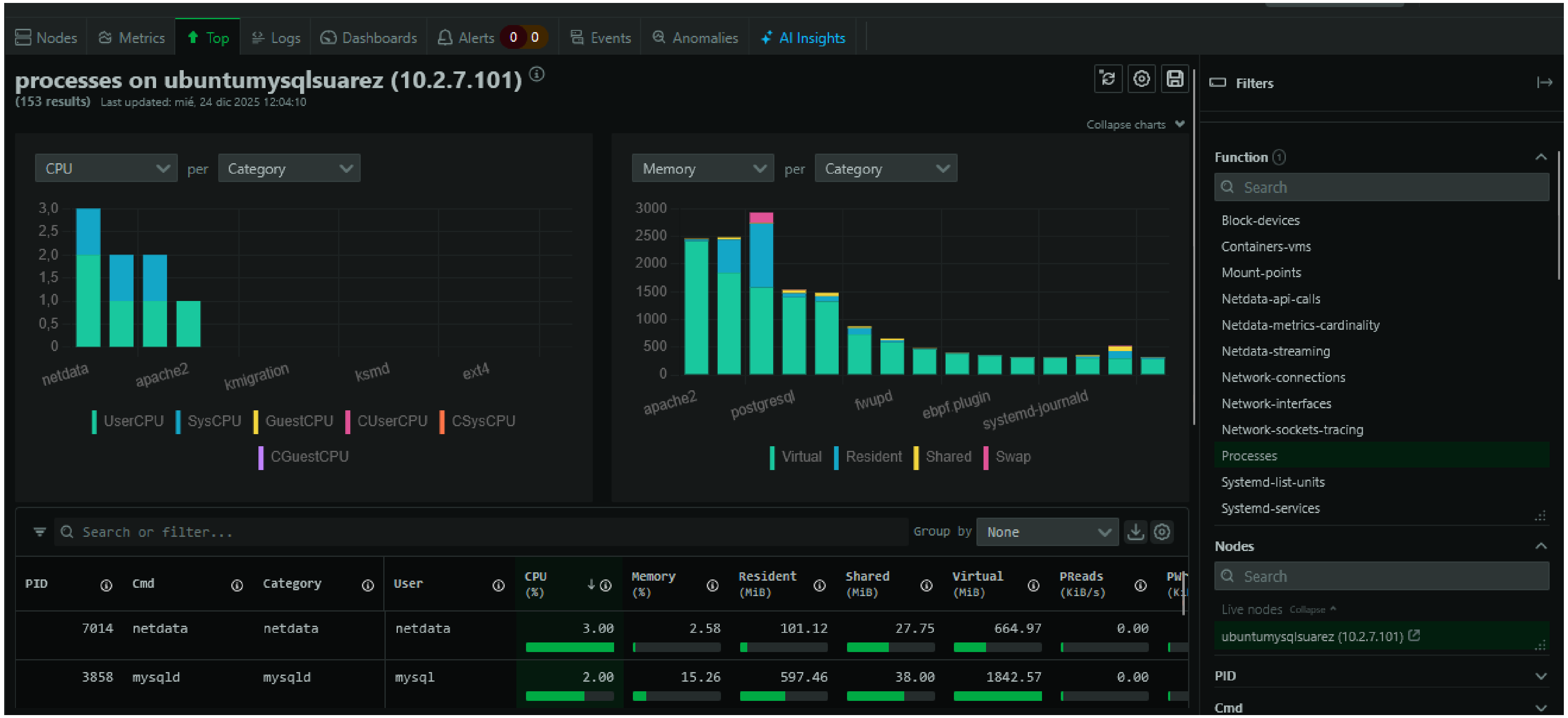

System > Compute > PSI > Memory > Some
---- Some Pressure
Memory some ... • system.memory_some_pressure • [percent]
Group by dimension the AVG() 1 node 1 system 3 dimensions 2 labels
each as AVG() every 4s
Reset
percent
1
0,8
0,6
0,4
0,2
0
11:30:00
11:35:00
11:40:00
Hovering: mié, 24 dic 2025 • 11:28:24
some 10 0 %
some 60 0 %
some 300 0 %
Expand - Chart Analysis
M • system.memory_some_pressure_stall_time • [milliseconds]
Group by dimension the AVG() 1 node 1 system 1 dimension 2 labels
each as AVG() every 4s
milliseconds
4
3
2
1
0
time 0 ms
---- Full Pressure
Memory full pres... • system.memory_full_pressure • [percent]
M... • system.memory_full_pressure_stall_time • [milliseconds]
mié, 24 dic 2025 • 11:28:24

Diagrama de uso de procesos, de aquellos pertenecientes al sistema, etc.

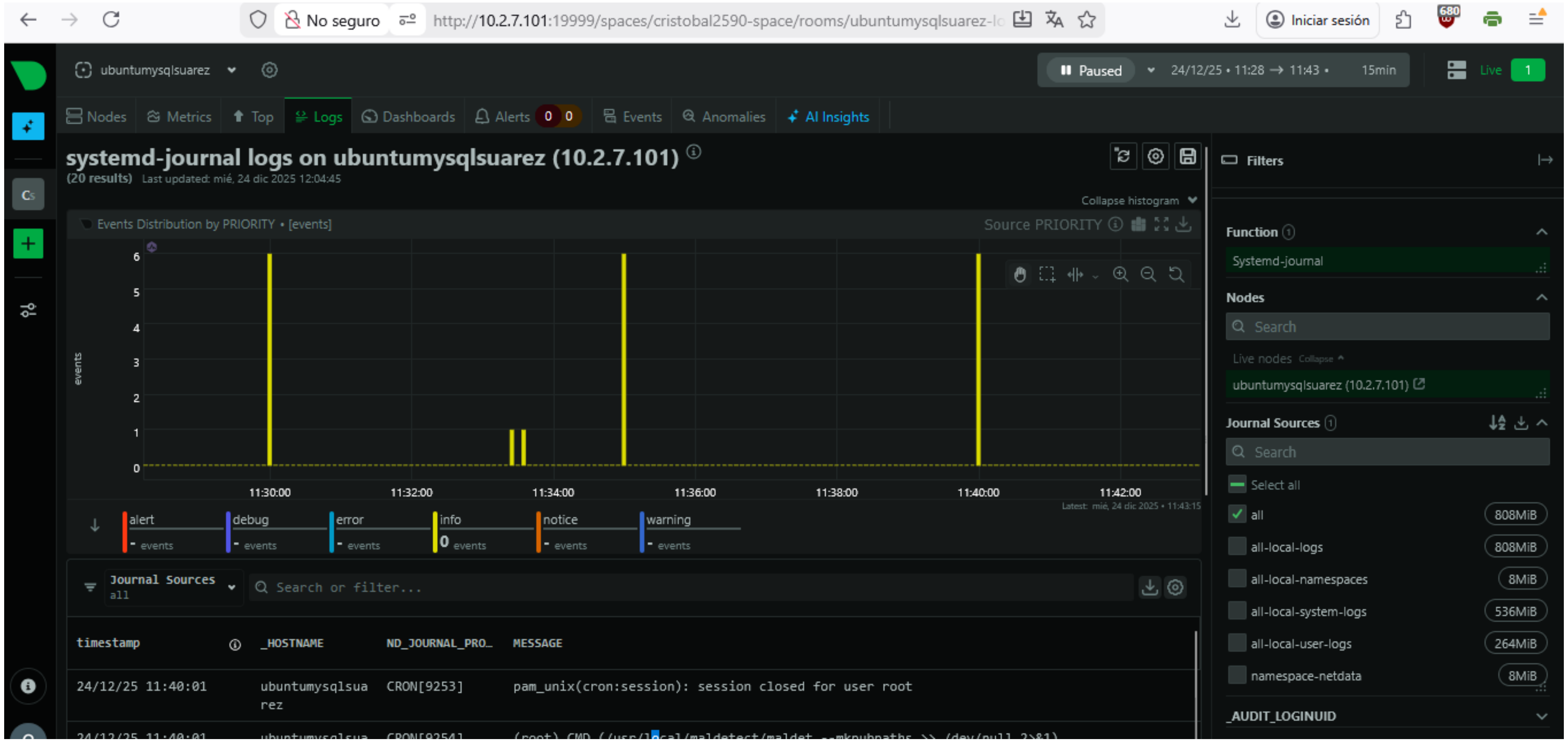


processes on ubuntumysqlsuarez (10.2.7.101)

(153 results) Last updated: mié, 24 dic 2025 12:04:10

PID	Cmd	Category	User	CPU (%)	Memory (%)	Resident (MiB)	Shared (MiB)	Virtual (MiB)	PReads (KiB/s)	PWr (Ki
7014	netdata	netdata	netdata	3.00	2.58	101.12	27.75	664.97	0.00	
3858	mysqld	mysqld	mysql	2.00	15.26	597.46	38.00	1842.57	0.00	

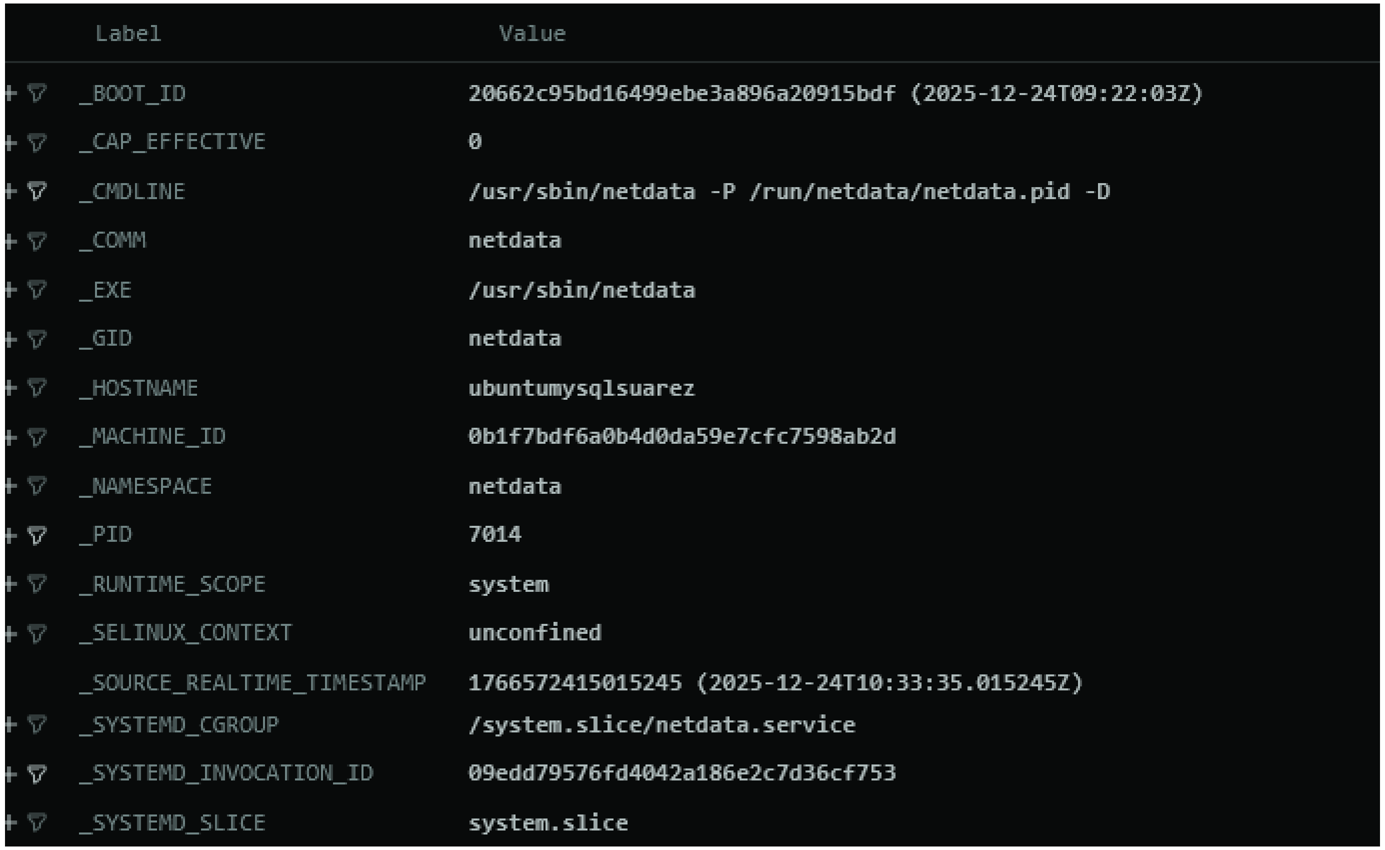
Creación de logs:



systemd-journal logs on ubuntumysqlsuarez (10.2.7.101)
(20 results) Last updated: mié, 24 dic 2025 12:04:45

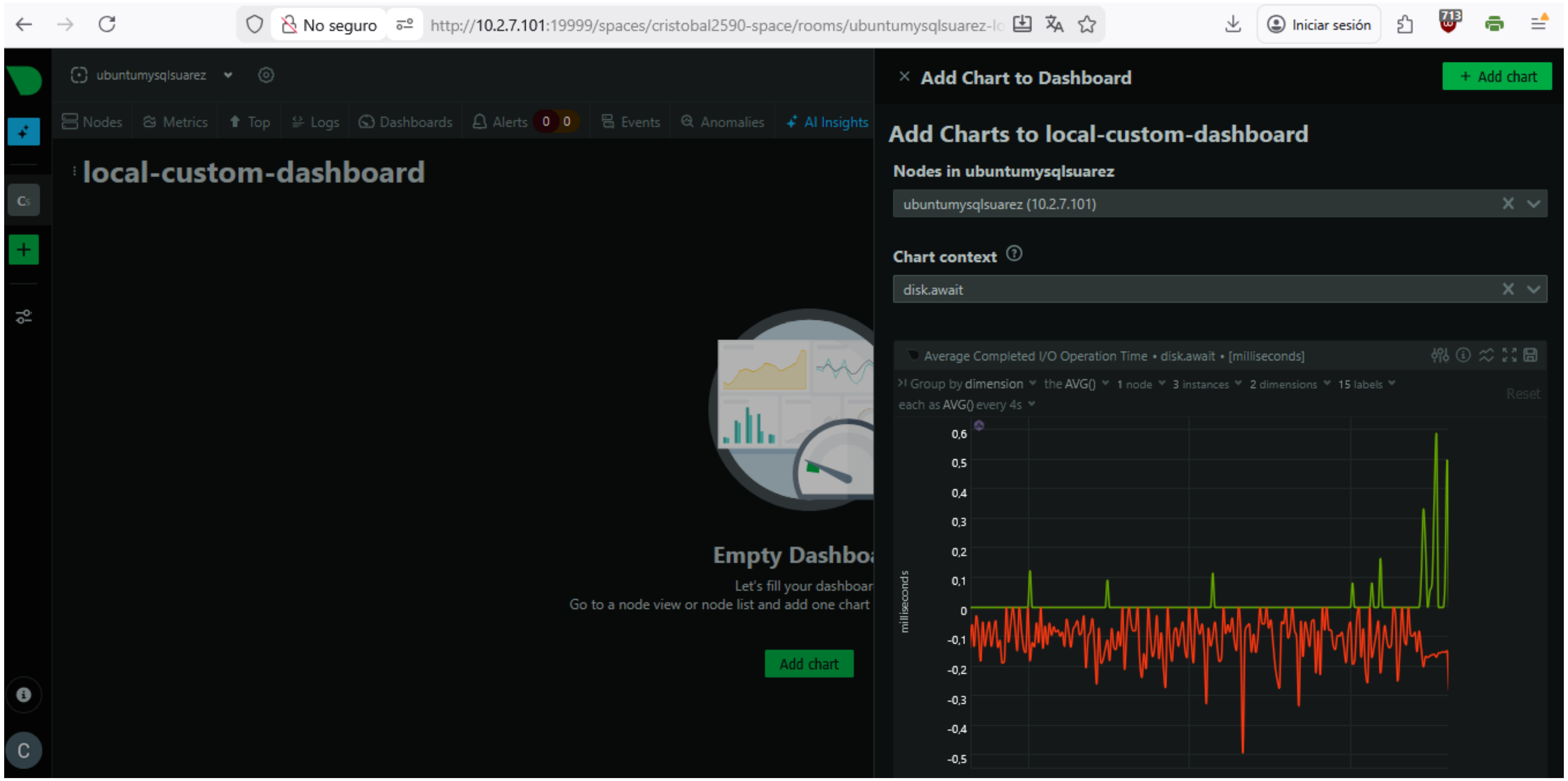
```
timestamp           _HOSTNAME        ND_JOURNAL_PRO_   MESSAGE
24/12/25 11:40:01   ubuntumysqlsua   CRON[9253]        pam_unix(cron:session): session closed for user root
                    rez
24/12/25 11:40:01   ubuntumysqlsua   CRON[9254]        (root) CMD (/usr/local/maldetect/maldet --mkpubpaths >> /dev/null 2>&1)
```

Podemos ver el contenido de los logs:

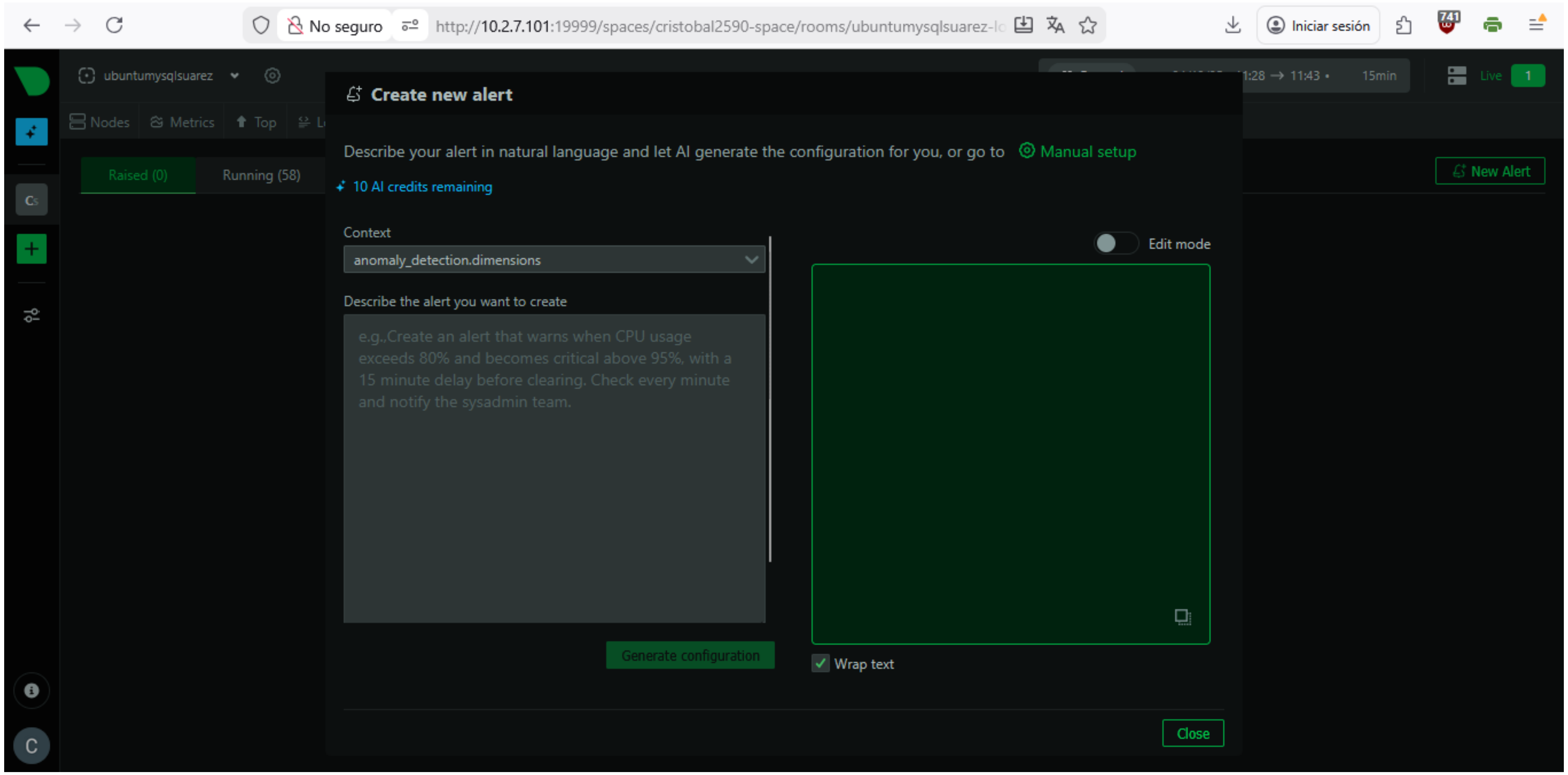


```
Label                         Value
_BOOT_ID                      20662c95bd16499ebe3a896a20915bdf (2025-12-24T09:22:03Z)
_CAP_EFFECTIVE                0
_CMDLINE                      /usr/sbin/netdata -P /run/netdata/netdata.pid -D
_COMM                         netdata
_EXE                          /usr/sbin/netdata
_GID                          netdata
_HOSTNAME                     ubuntumysqlsuarez
_MACHINE_ID                   0b1f7bdf6a0b4d0da59e7cfc7598ab2d
_NAMESPACE                    netdata
_PID                          7014
_RUNTIME_SCOPE                system
_SELINUX_CONTEXT              unconfined
_SOURCE_REALTIME_TIMESTAMP    1766572415015245 (2025-12-24T10:33:35.015245Z)
_SYSTEMD_CGROUP               /system.slice/netdata.service
_SYSTEMD_INVOCATION_ID        09edd79576fd4042a186e2c7d36cf753
_SYSTEMD_SLICE                system.slice
```

Podemos crear un Dashboard personalizado:



Generar nuestras propias alertas:



http://10.2.7.101:19999/spaces/cristobal2590-space/rooms/ubuntumysqlsuarez-lo

Create new alert

Describe your alert in natural language and let AI generate the configuration for you, or go to Manual setup

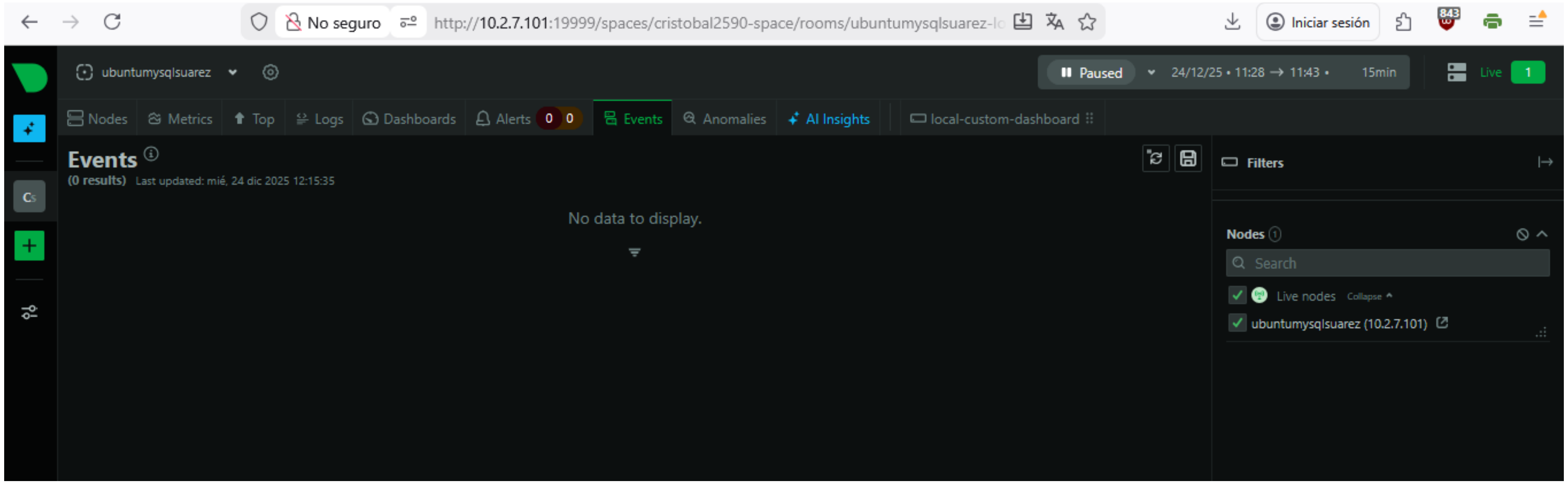
10 AI credits remaining

Context: anomaly_detection.dimensions

Describe the alert you want to create

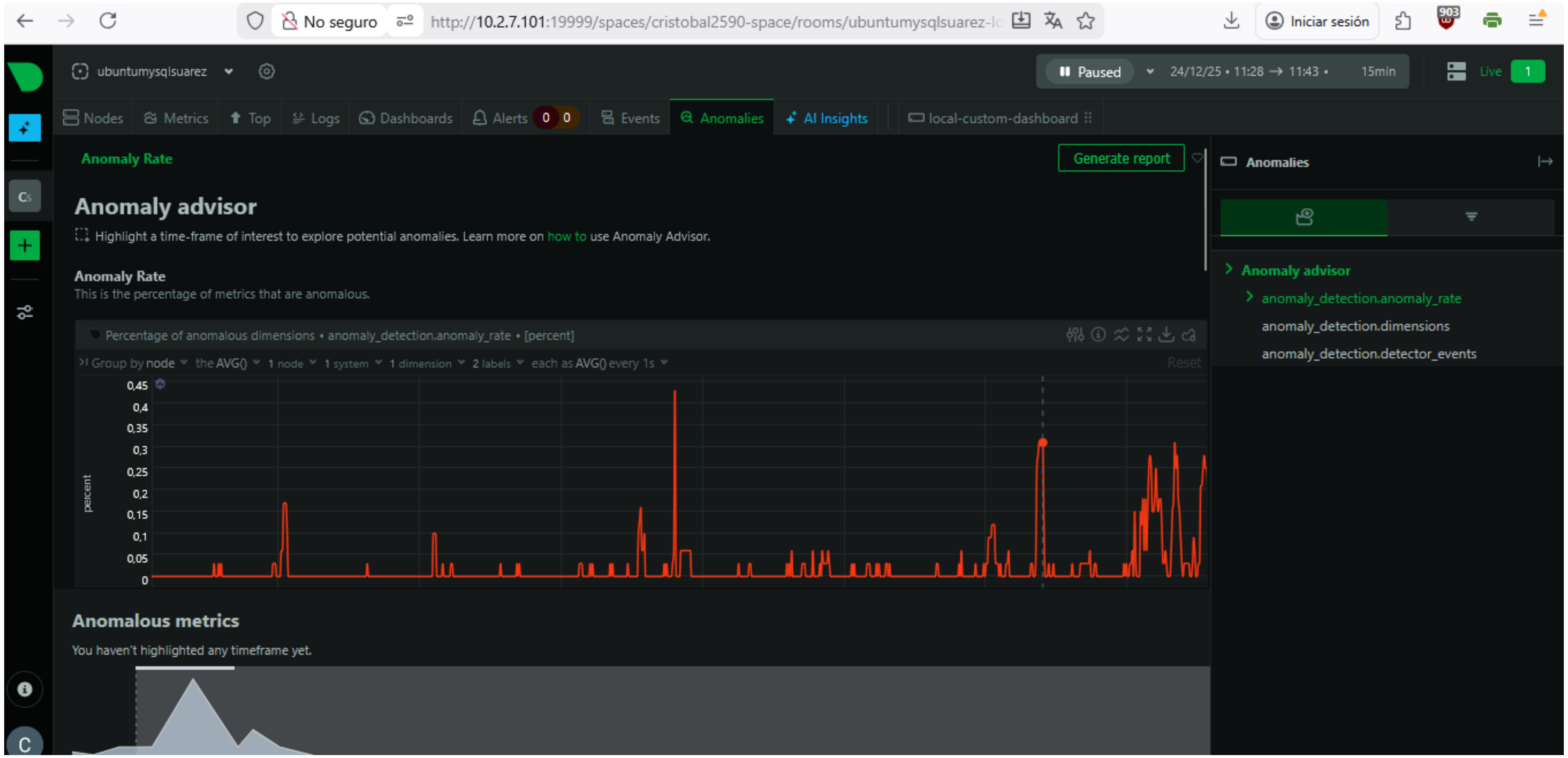
e.g.,Create an alert that warns when CPU usage exceeds 80% and becomes critical above 95%, with a 15 minute delay before clearing. Check every minute and notify the sysadmin team.

Donde se guardarán los eventos:



Events
(0 results) Last updated: mié, 24 dic 2025 12:15:35
No data to display.

Detector de anomalías:



Anomaly advisor

Highlight a time-frame of interest to explore potential anomalies. Learn more on how to use Anomaly Advisor.

Anomaly Rate

This is the percentage of metrics that are anomalous.

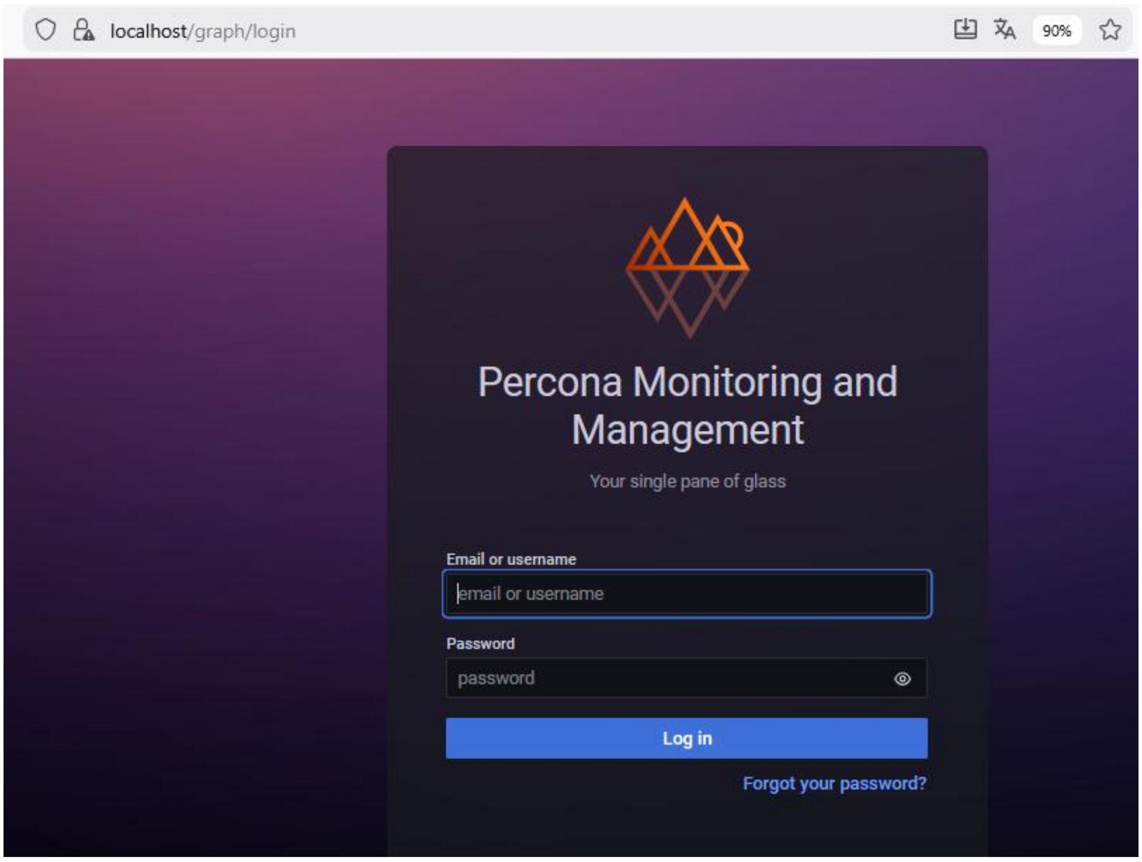
Percentage of anomalous dimensions • anomaly_detection.anomaly_rate • [percent]

Anomalous metrics

You haven't highlighted any timeframe yet.

5) Monitorización con PMM (Percona Monitoring and Management). Usando la imagen contenida en el docker compose.

https://localhost:443



localhost/graph/login

Percona Monitoring and Management

Your single pane of glass

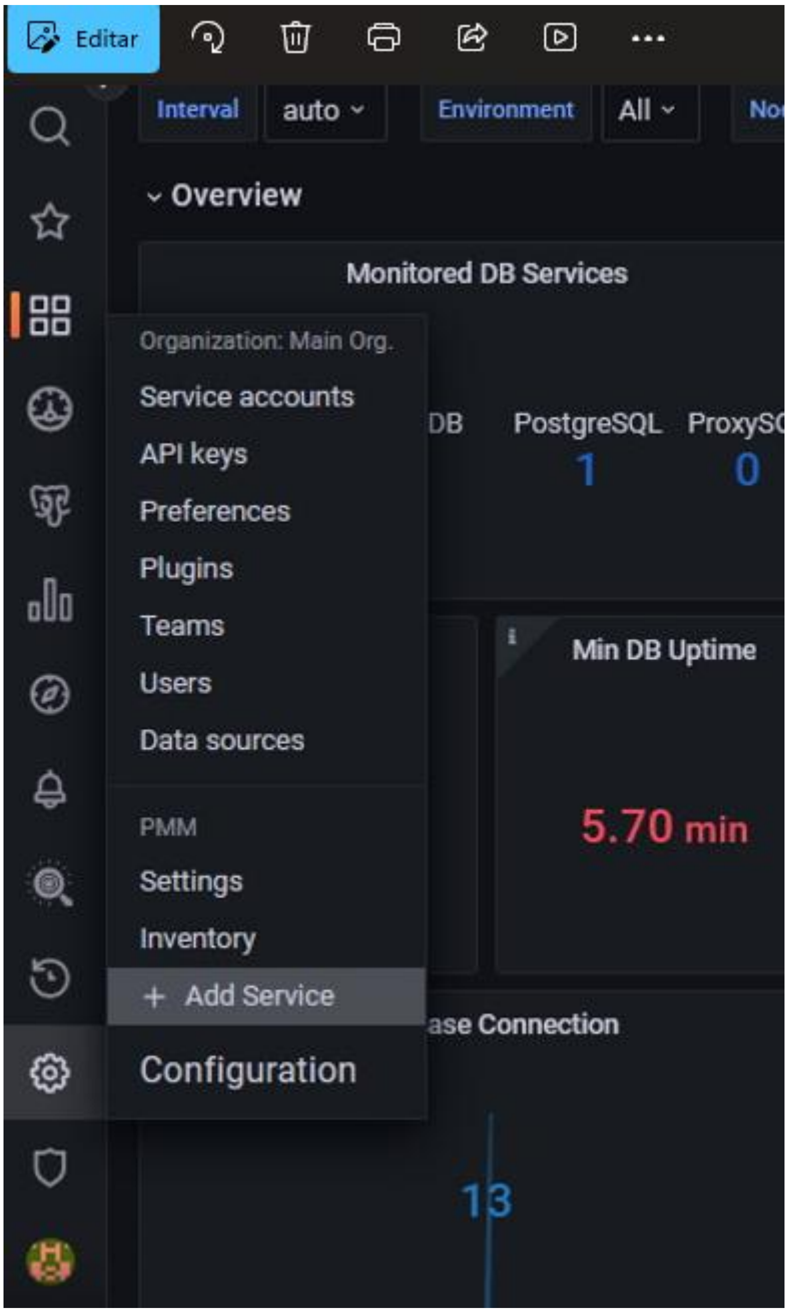
Email or username: email or username

Password: password

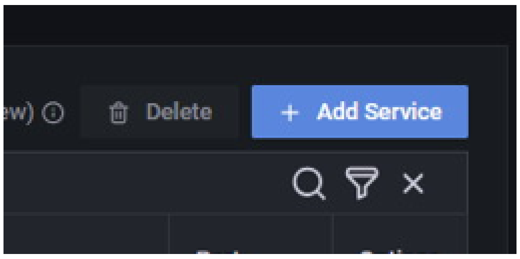
Por defecto el usuario es “admin” y la contraseña “admin”. La primera vez que iniciemos nos pedirá establecer una nueva contraseña.

Conecta MySQL al PMM

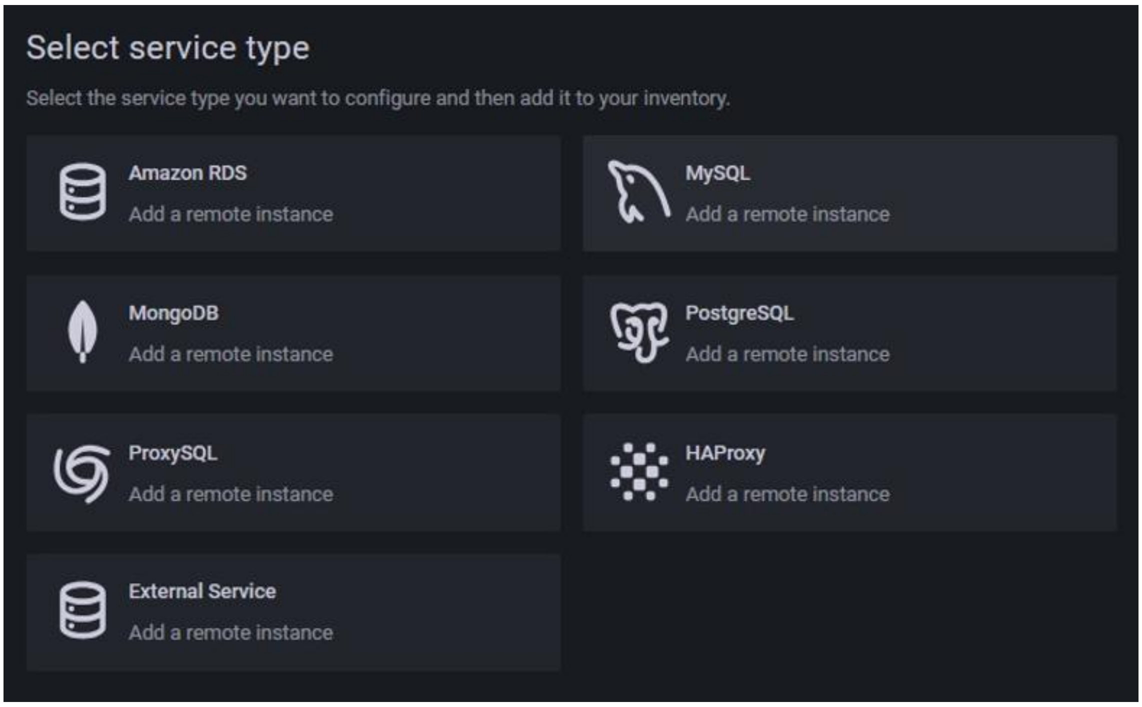
Panel izquierdo →Configuración



Zona derecha: Añadir Servicio.



Elegimos el tipo de servicio, en nuestro caso "MySQL":



Select service type

Select the service type you want to configure and then add it to your inventory.

Amazon RDS
Add a remote instance

MySQL
Add a remote instance

MongoDB
Add a remote instance

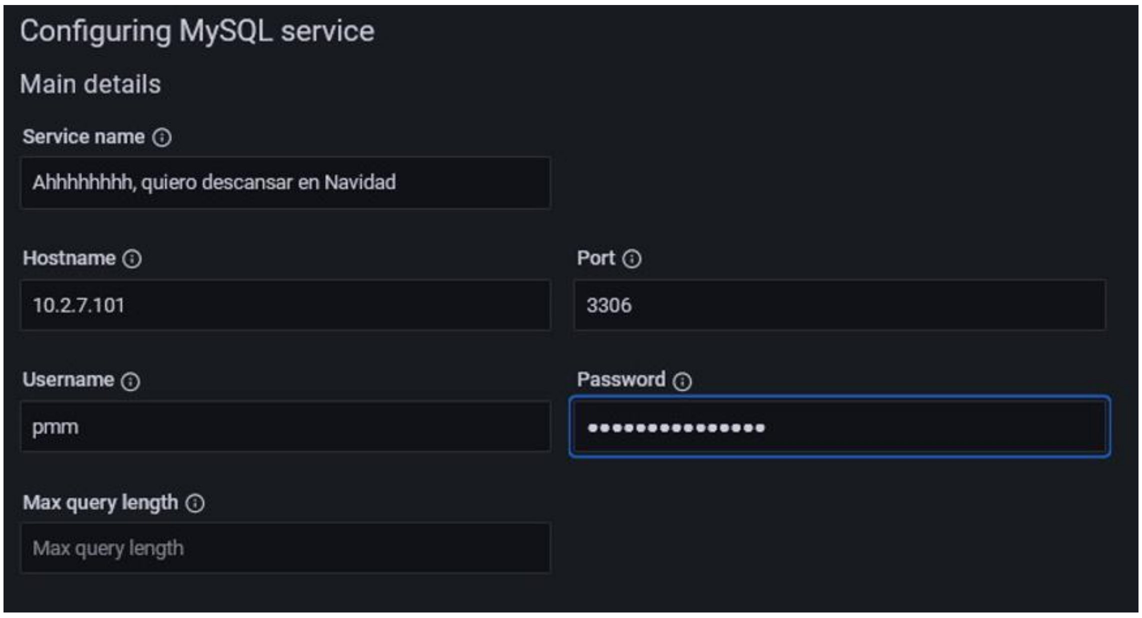
PostgreSQL
Add a remote instance

ProxySQL
Add a remote instance

HAProxy
Add a remote instance

External Service
Add a remote instance

Introducimos los datos necesarios:



Configuring MySQL service

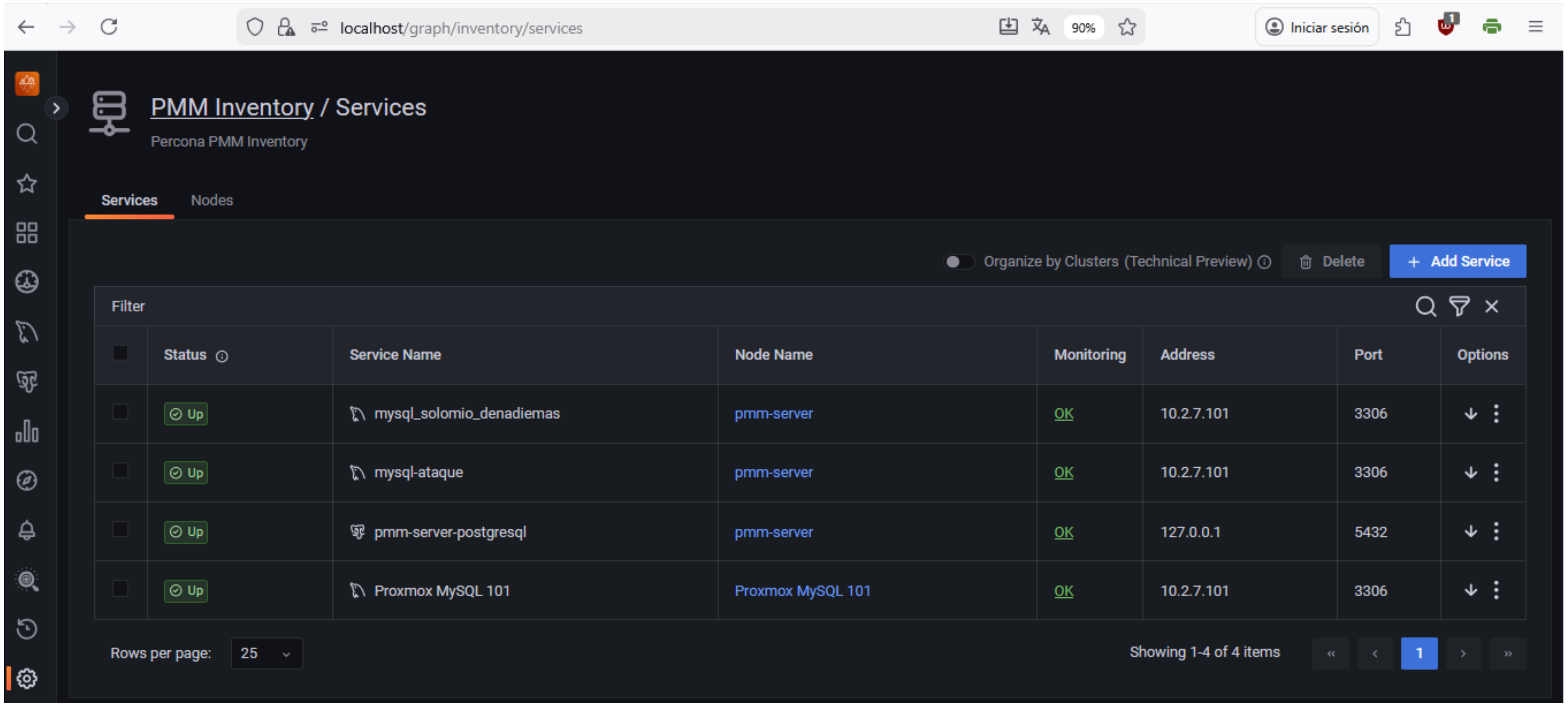
Main details

Service name	Ahhhhhhhh, quiero descansar en Navidad
Hostname	10.2.7.101
Port	3306
Username	pmm
Password	••••••••••••••••
Max query length	Max query length

También podemos usar el siguiente comando desde el host donde tenemos el contenedor Docker.

docker exec -it pmm-server2 pmm-admin add mysql --server-url=https://admin:admin@127.0.0.1:443 --server-insecure-tls --username=pmm --password=pass_segura_pmm --host=10.2.7.101 --port=3306 --query-source=perfschema mysql-ataque

Servicios creados:

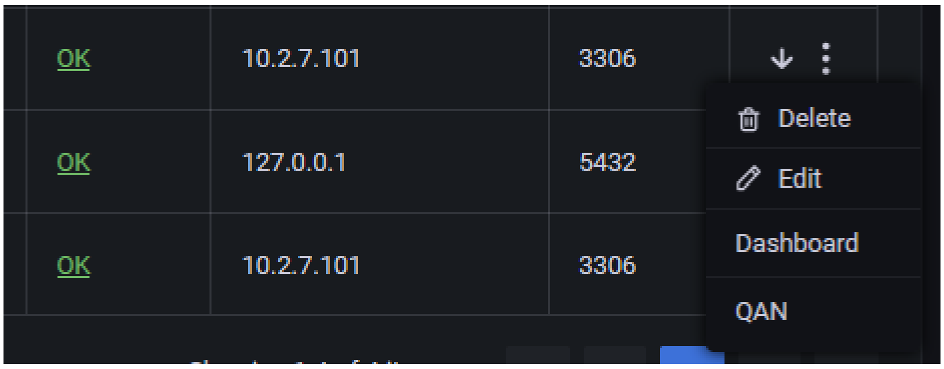


Status	Service Name	Node Name	Monitoring	Address	Port	Options
Up	mysql_solomio_denadiemas	pmm-server	OK	10.2.7.101	3306	
Up	mysql-ataque	pmm-server	OK	10.2.7.101	3306	
Up	pmm-server-postgresql	pmm-server	OK	127.0.0.1	5432	
Up	Proxmox MySQL 101	Proxmox MySQL 101	OK	10.2.7.101	3306	

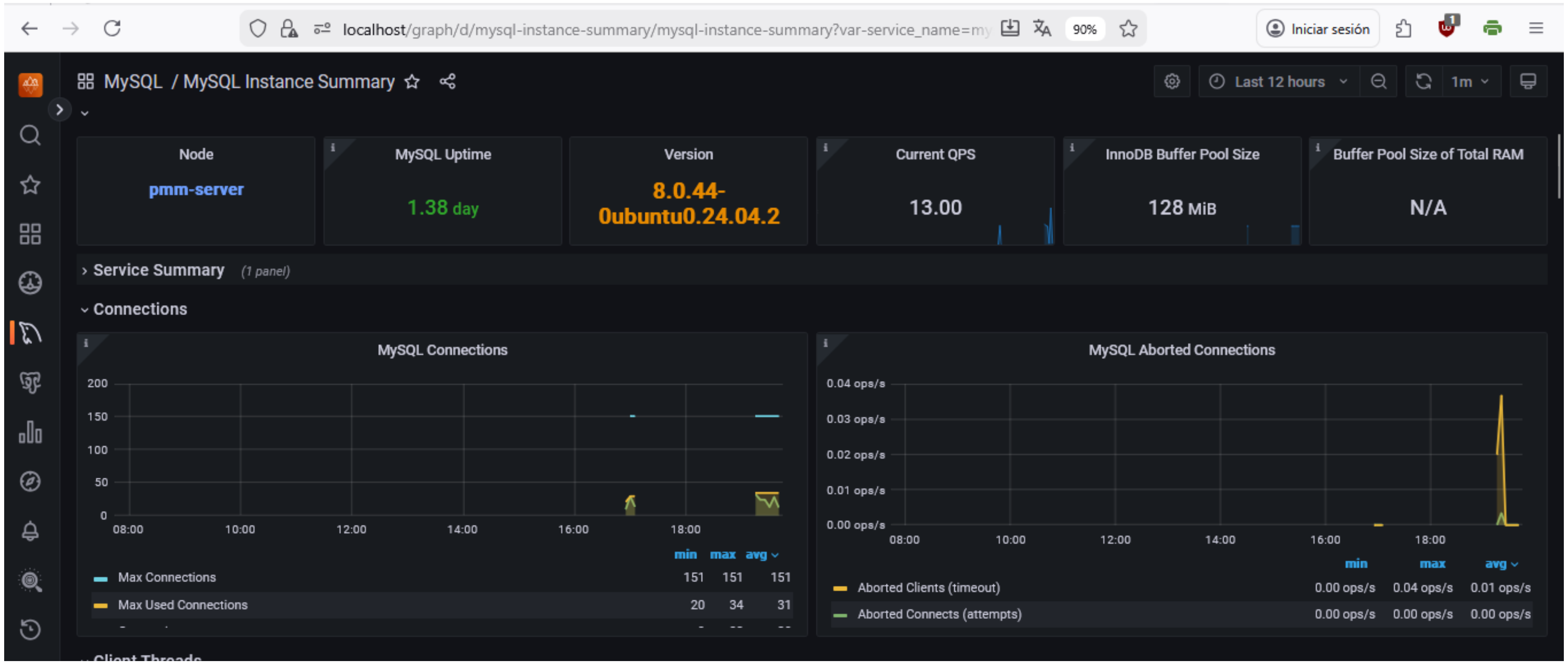
Rows per page: 25

Showing 1-4 of 4 items

Para ir al dashboard:



OK	10.2.7.101	3306
OK	127.0.0.1	5432
OK	10.2.7.101	3306


localhost/graph/d/mysql-instance-summary/mysql-instance-summary?var-service_name=my
90%
Iniciar sesión
MySQL / MySQL Instance Summary
Last 12 hours
1m

Node	MySQL Uptime	Version	Current QPS	InnoDB Buffer Pool Size	Buffer Pool Size of Total RAM
pmm-server	1.38 day	8.0.44-0ubuntu0.24.04.2	13.00	128 MiB	N/A

› Service Summary (1 panel)

⌄ Connections

MySQL Connections

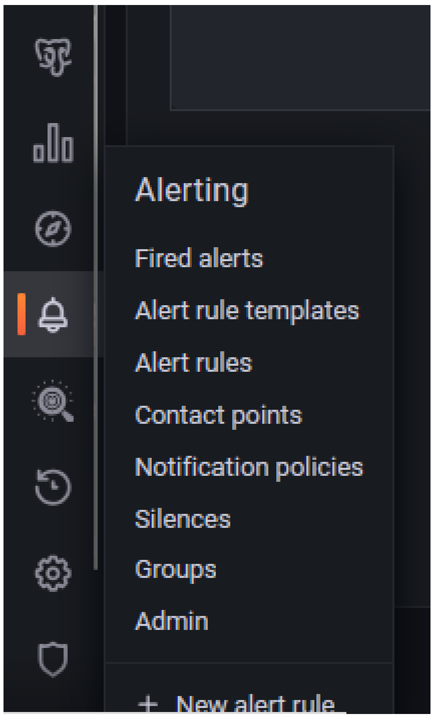
	min	max	avg
Max Connections	151	151	151
Max Used Connections	20	34	31

MySQL Aborted Connections

	min	max	avg
Aborted Clients (timeout)	0.00 ops/s	0.04 ops/s	0.01 ops/s
Aborted Connects (attempts)	0.00 ops/s	0.00 ops/s	0.00 ops/s

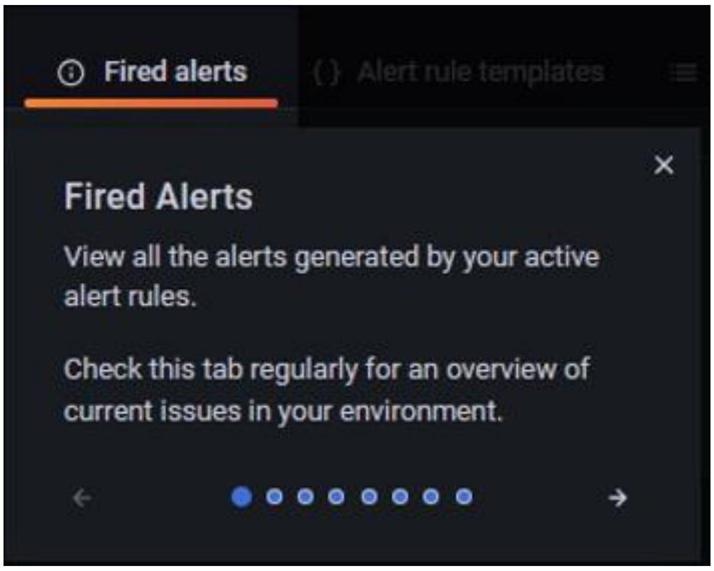
Investiga, prueba y documenta las opciones que tenga que ver con generación de alertas.

En el panel de la izquierda:



- Fired Alerts:

Nos muestra aquellas alertas que han saltado.



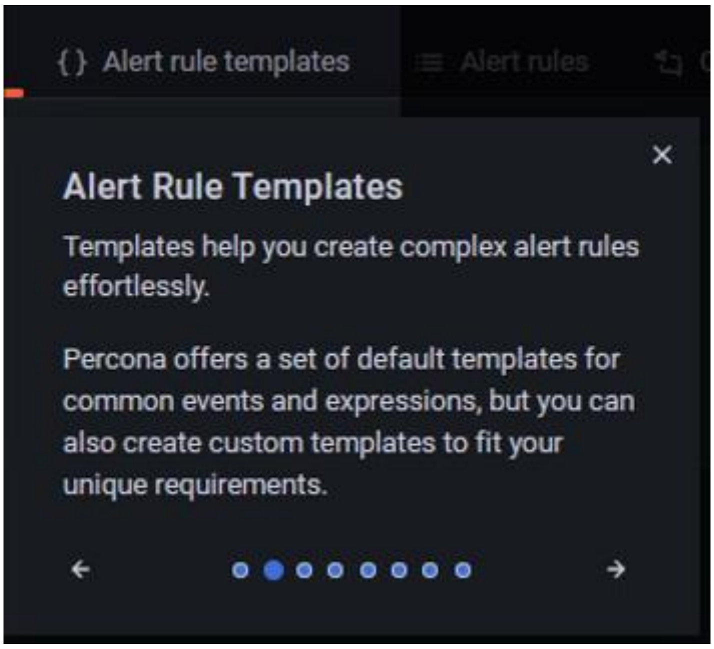
Fired Alerts

View all the alerts generated by your active alert rules.

Check this tab regularly for an overview of current issues in your environment.

- Plantillas de alertas:

Tenemos a nuestra disposición plantillas para los eventos y situaciones más comunes.



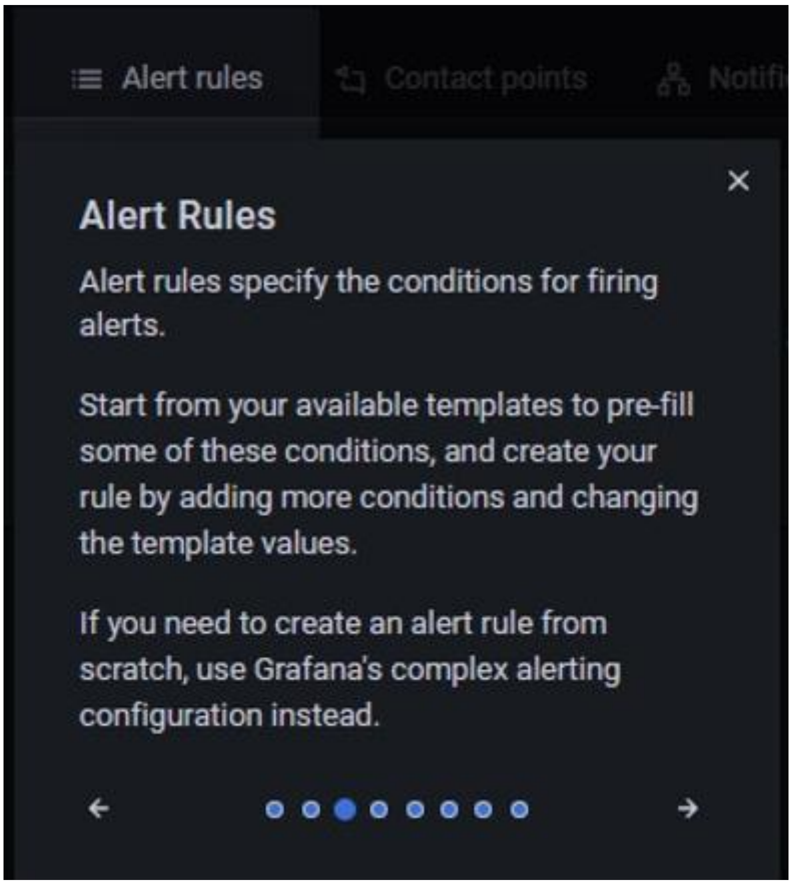
Alert Rule Templates

Templates help you create complex alert rules effortlessly.

Percona offers a set of default templates for common events and expressions, but you can also create custom templates to fit your unique requirements.

- Reglas de Alerta:

Especificamos las condiciones que hacen saltar las alertas.



Alert Rules

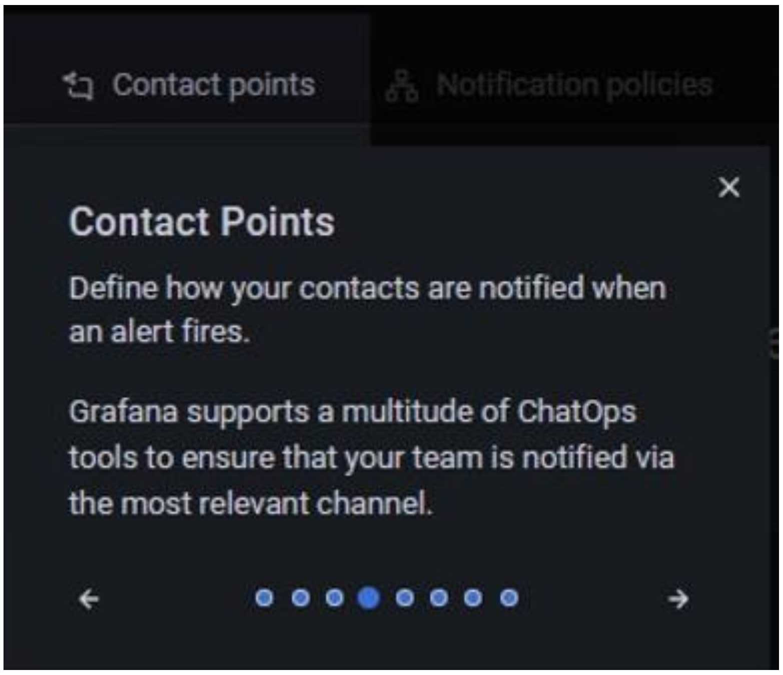
Alert rules specify the conditions for firing alerts.

Start from your available templates to pre-fill some of these conditions, and create your rule by adding more conditions and changing the template values.

If you need to create an alert rule from scratch, use Grafana's complex alerting configuration instead.

- Contact Points:

Establece la forma de contactar para avisar de las alertas.

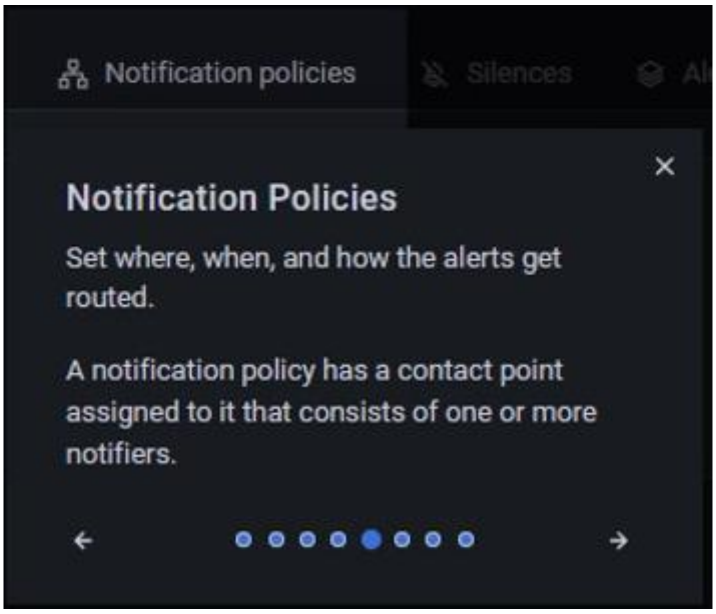


Contact Points

Define how your contacts are notified when an alert fires.

Grafana supports a multitude of ChatOps tools to ensure that your team is notified via the most relevant channel.

- Políticas de notificación:



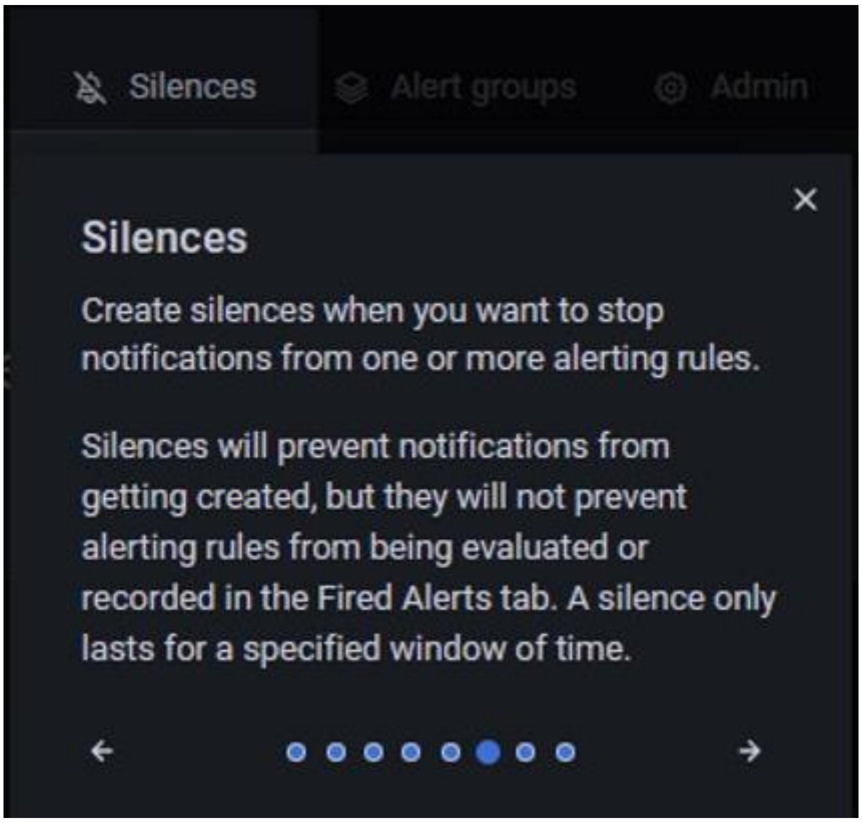
Notification Policies

Set where, when, and how the alerts get routed.

A notification policy has a contact point assigned to it that consists of one or more notifiers.

- Silencios:

Podemos establecer periodos de "silencio" en los que no se mandarán notificaciones.



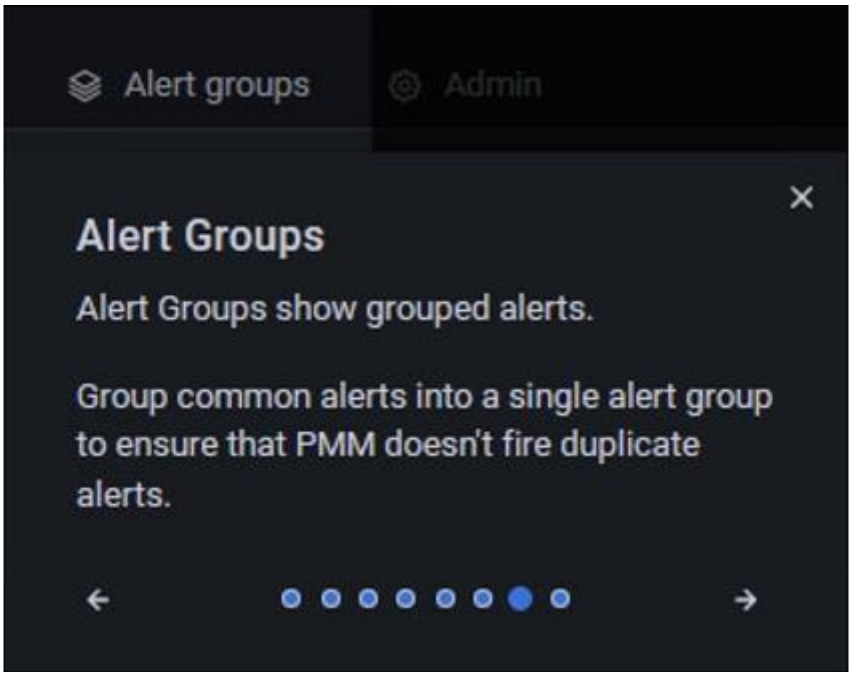
Silences

Create silences when you want to stop notifications from one or more alerting rules.

Silences will prevent notifications from getting created, but they will not prevent alerting rules from being evaluated or recorded in the Fired Alerts tab. A silence only lasts for a specified window of time.

- Grupos de Alertas:

Podemos agrupar las alertas.



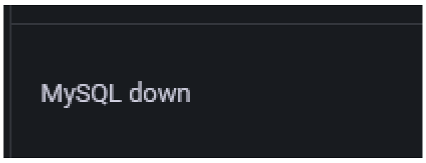
Alert Groups

Alert Groups show grouped alerts.

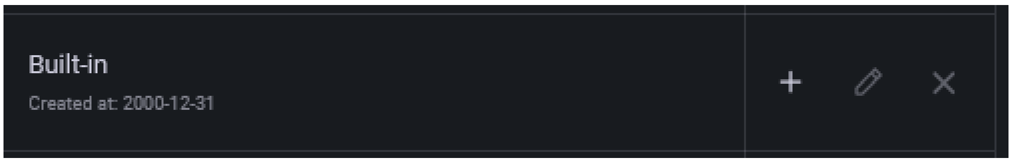
Group common alerts into a single alert group to ensure that PMM doesn't fire duplicate alerts.

Prueba 1: Vamos a usar una de las plantillas.

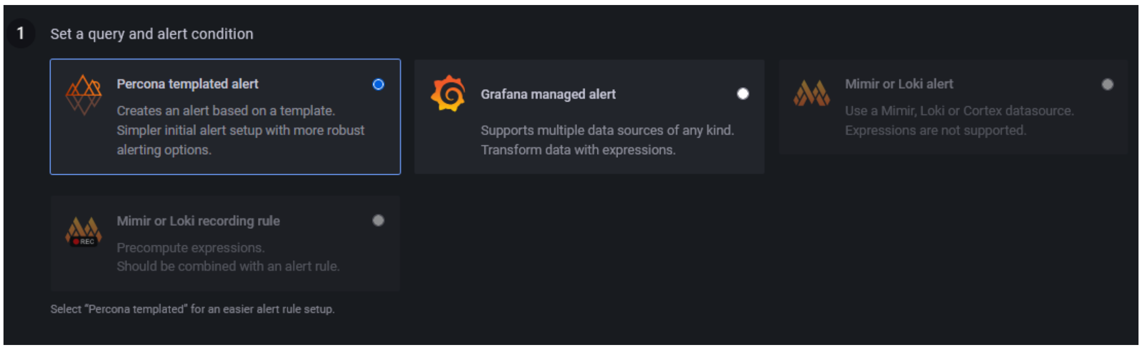
En nuestro caso elegimos la de “**MySQL Down**” que nos avisa cuando el servicio del servidor está caído.


MySQL down

Vamos a “Alert rule templates” y nos vamos a la sección de la derecha y le damos a la cruz para añadirlo.


Built-in
Created at: 2000-12-31

En nuestro caso usamos el Percona.


1 Set a query and alert condition

Percona templated alert
Creates an alert based on a template. Simpler initial alert setup with more robust alerting options.

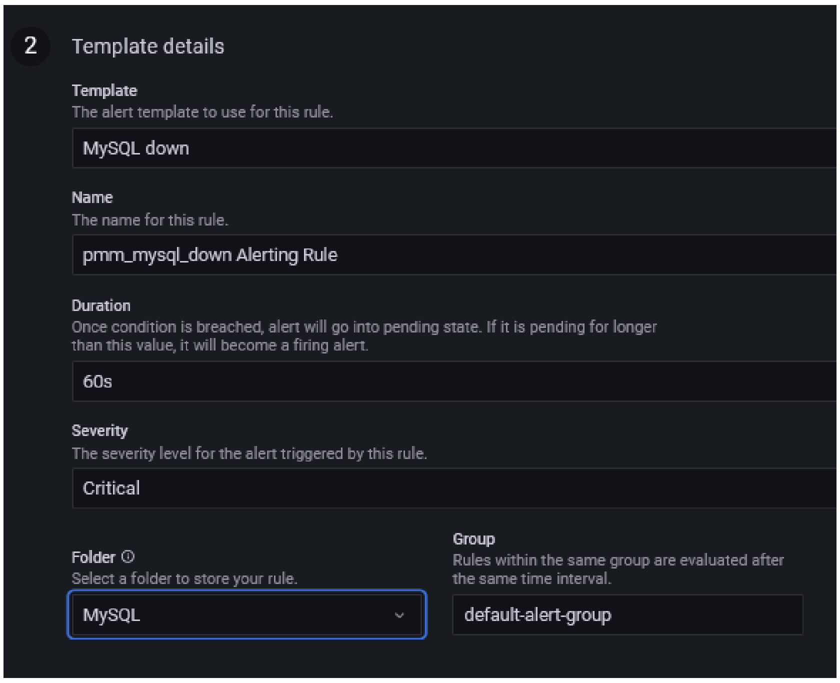
Grafana managed alert
Supports multiple data sources of any kind. Transform data with expressions.

Mimir or Loki alert
Use a Mimir, Loki or Cortex datasource. Expressions are not supported.

Mimir or Loki recording rule
Precompute expressions. Should be combined with an alert rule.

Select "Percona templated" for an easier alert rule setup.

Rellenamos datos. Guardamos y salimos.


2 Template details

Template
The alert template to use for this rule.
MySQL down

Name
The name for this rule.
pmm_mysql_down Alerting Rule

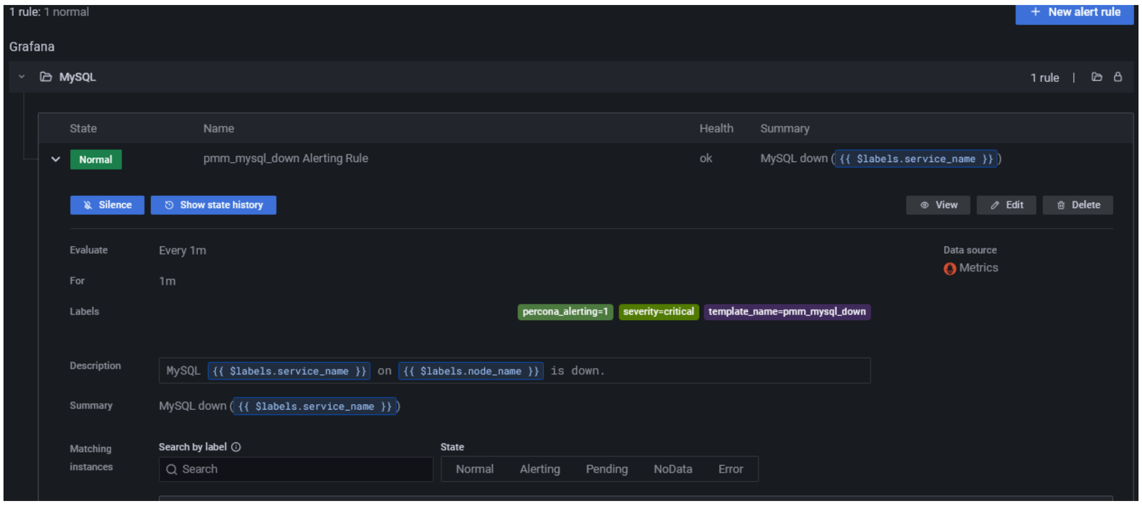
Duration
Once condition is breached, alert will go into pending state. If it is pending for longer than this value, it will become a firing alert.
60s

Severity
The severity level for the alert triggered by this rule.
Critical

Folder
Select a folder to store your rule.
MySQL

Group
Rules within the same group are evaluated after the same time interval.
default-alert-group

Podremos ver en “Alert Rules” que se ha guardado.



State	Name	Health	Summary
Normal	pmm_mysql_down Alerting Rule	ok	MySQL down ({{ $labels.service_name }})

Evaluate Every 1m

For 1m

Labels percona_alerting=1 severity=critical template_name=pmm_mysql_down

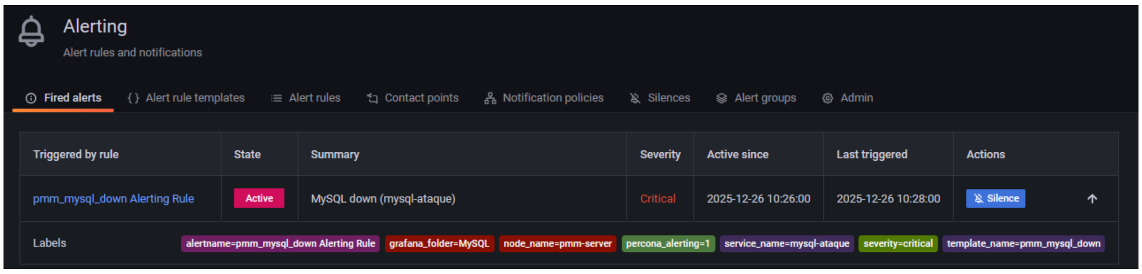
Data source Metrics

Description `MySQL {{ $labels.service_name }} on {{ $labels.node_name }} is down.`

Summary MySQL down ({{ $labels.service_name }})

Matching instances — Search by label: Search — State: Normal Alerting Pending NoData Error

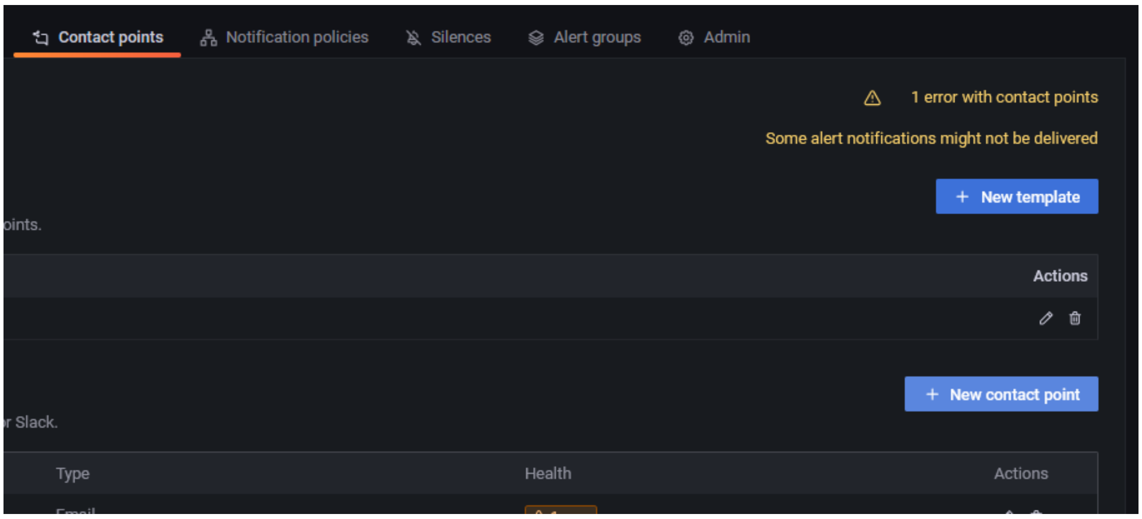
Para comprobar que funciona, en este caso apagamos el servidor. En nuestro caso puede tardar un poco porque establecimos que saltase la alarma si la condición se mantenía por 60 segundos.



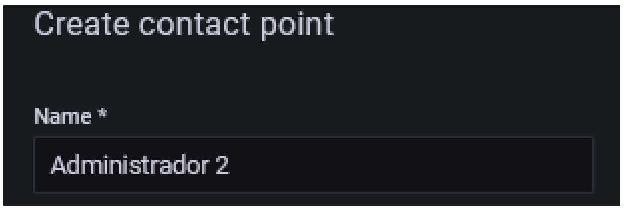
Triggered by rule	State	Summary	Severity	Active since	Last triggered	Actions
pmm_mysql_down Alerting Rule	Active	MySQL down (mysql-ataque)	Critical	2025-12-26 10:26:00	2025-12-26 10:28:00	Silence

Labels alertname=pmm_mysql_down Alerting Rule grafana_folder=MySQL node_name=pmm-server percona_alerting=1 service_name=mysql-ataque severity=critical template_name=pmm_mysql_down

Añadir Contacto: Para poder enviar mensajes de correo con las alertas.

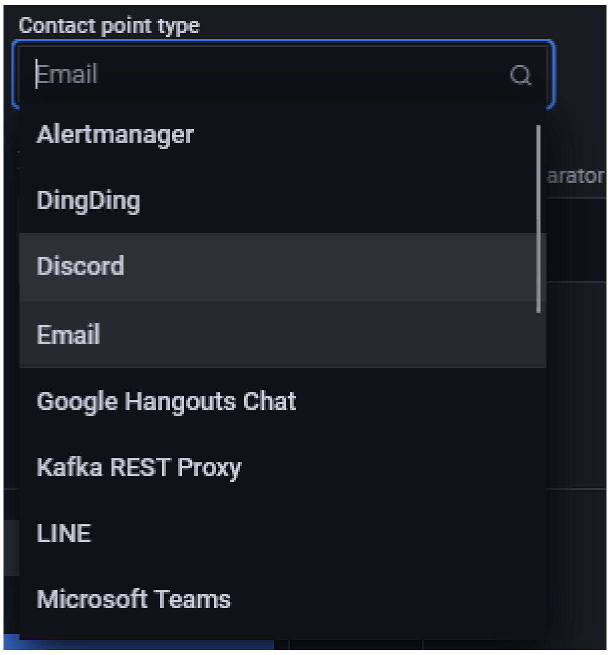


Establecer un nombre (podemos poner lo que queremos).

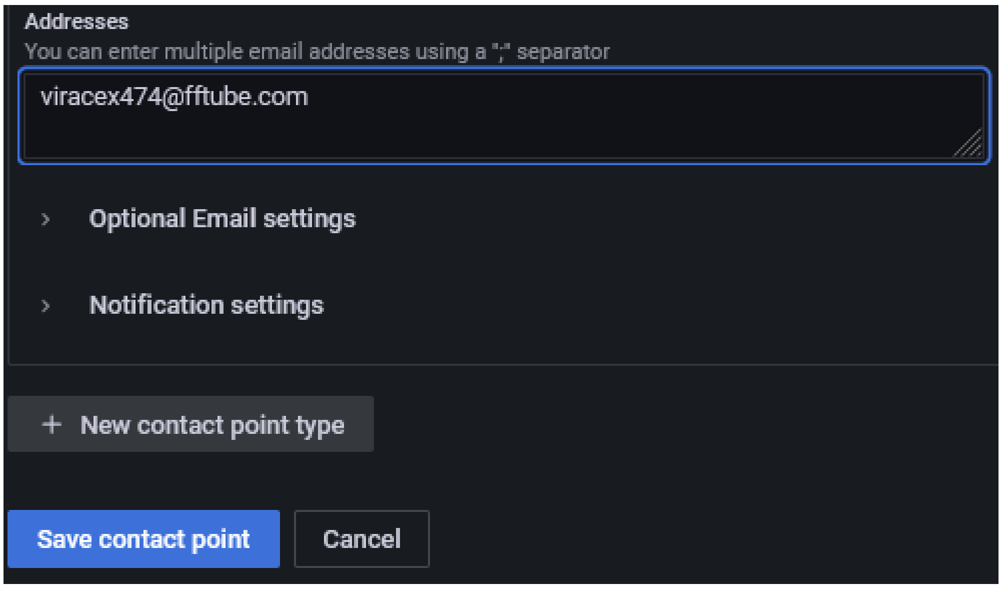


Name *
Administrador 2

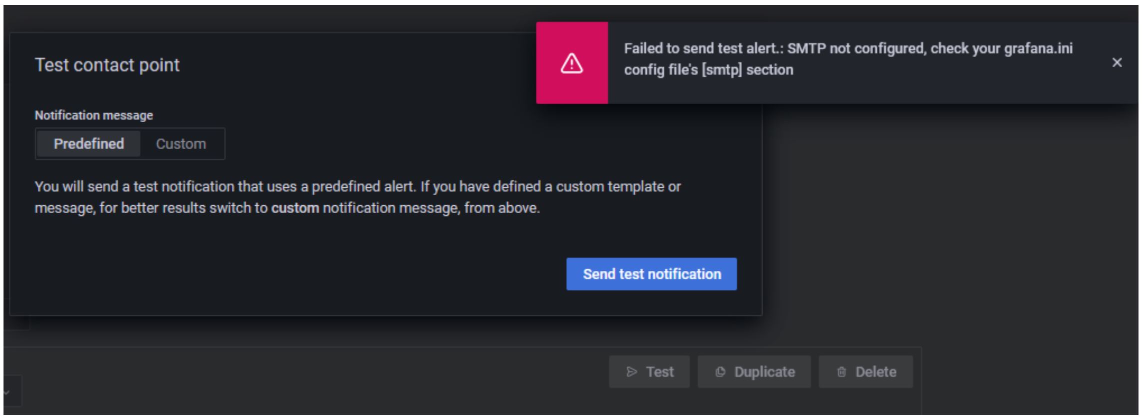
Tipo de contacto. En nuestro caso usamos email.



Introducimos el correo electrónico y guardamos.


Addresses
You can enter multiple email addresses using a ";" separator
viracex474@fftube.com

ATENCIÓN: Para poder utilizar esta opción será necesario configurar Grafana para usar un servidor de correo (SMPT).


Test contact point
Notification message
Predefined Custom
You will send a test notification that uses a predefined alert. If you have defined a custom template or message, for better results switch to **custom** notification message, from above.
Failed to send test alert.: SMTP not configured, check your grafana.ini config file's [smtp] section