

Installation et configuration de Prometheus et Grafana

**Prometheus** est un système de surveillance open source conçu pour collecter et stocker des métriques de performance et de santé des applications. Il est largement utilisé dans le domaine du cloud computing pour la surveillance et l'alerte en temps réel, offrant une visualisation des données et des fonctionnalités avancées telles que l'agrégation, la requête et l'alerte basée sur les métriques collectées.

**Grafana** est une plateforme open source de visualisation et d'analyse de données. Elle permet de créer des tableaux de bord interactifs et personnalisables pour surveiller et analyser diverses sources de données, telles que les métriques, les journaux et les bases de données. Grafana offre une large gamme de fonctionnalités de visualisation, de filtrage, de recherche et de collaboration, en faisant un outil populaire pour la surveillance et l'analyse des données.

### ***Machine : Ubuntu 20.04***

#### **Etape1 : Installation de Prometheus**

Wget<https://github.com/prometheus/prometheus/releases/download/v2.18.1/prometheus-2.18.1.linux-amd64.tar.gz>: permet de télécharger le dossier zipper de Prometheus

**tar -xzf prometheus-2.18.1.linux-amd64.tar.gz** : permet de dézipper le dossier de Prometheus téléchargé précédemment.

**sudo useradd -s /sbin/false prometheus**: permet de créer l'utilisateur prometheus

Cet utilisateur sera utilisé lorsqu'on veut effectuer des modifications au niveau du fichier de configuration de prometheus

**Cd /etc** : permet d'accéder au dossier etc du système d'exploitation

**Mkdir prometheus** : crée le dossier prometheus

**Cp -r prometheus-2.18.1.linux-amd64 /etc/prometheus** : permet de copier le contenu du dossier de prometheus téléchargé vers le nouveau dossier prometheus créé

**sudo chmod 755 /etc/prometheus -R** : permet de donner d'exécution sur ce dossier à l'utilisateur Prometheus.

**sudo chown prometheus:prometheus /etc/prometheus -R** : ajoute de l'utilisateur aux groupe prometheus.

```

[jilmonde-admin@vm01: /etc/prometheus
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage
* Introducing Expanded Security Maintenance for Applications.
  Receive updates to over 25,000 software packages with your
  Ubuntu Pro subscription. Free for personal use.

https://ubuntu.com/pro
Welcome!

This server is hosted by OPEN SOLUTIONS DIGITALES. If you have any questions or need help,
please don't hesitate to contact us at support@open.bj.

Last login: Tue Jul 18 02:01:31 2023 from 156.0.213.12
[jilmonde-admin@vm01: ~]$ cd /etc/systemd/system
[jilmonde-admin@vm01: /etc/systemd/system]$ sudo nano prometheus.service
[sudo] password for jilmonde-admin:
[jilmonde-admin@vm01: /etc/systemd/system]$ cd ..
[jilmonde-admin@vm01: /etc]$ cd prometheus/
[jilmonde-admin@vm01: /etc/prometheus]$ ls -l
total 28
drwxr-xr-x 2 prometheus prometheus 4096 Dec 17 2021 console_libraries
drwxr-xr-x 2 prometheus prometheus 4096 Dec 17 2021 consoles
-rw-r--r-- 1 root      root      171 Jun 28 18:09 dead-mans-snitch-rule.yml
-rw-r--r-- 1 prometheus prometheus 1757 Jul  5 12:22 prometheus.yml
-rw-r--r-- 1 root      root      313 Jul  9 00:00 ruleApch.yml
drwxr-xr-x 2 root      root      4096 Jul  3 15:40 rules
-rw-r--r-- 1 root      root      263 Jul  4 16:02 rule.yml
[jilmonde-admin@vm01: /etc/prometheus]$ 

```

Dans le dossier de prometheus nous avons le fichier **prometheus.yml**

Prometheus.yml est le fichier de configuration de prometheus.

**sudo nano /etc/systemd/system/prometheus.service** : cette commande permet de créer le fichier systemd de prometheus.

Les fichiers systemd sont utilisés pour la gestion du système et le démarrage des services sur les systèmes d'exploitation Linux.

Voici la configuration du fichier systemd

```

[jilmonde-admin@vm01: /etc/systemd/system
GNU nano 6.2
[Unit]
Description=Prometheus
Wants=network-online.target
After=network-online.target

StartLimitIntervalSec=500
StartLimitBurst=5

[Service]
User=prometheus
Group=prometheus
Type=simple
Restart=on-failure
RestartSec=5s
ExecStart=/usr/local/bin/prometheus \
--config.file=/etc/prometheus/prometheus.yml \
--storage.tsdb.path=/data \
--web.console.templates=/etc/prometheus/consoles \
--web.console.libraries=/etc/prometheus/console_libraries \
--web.listen-address=0.0.0.0:9090 \
--web.enable-lifecycle

[Install]
WantedBy=multi-user.target

```

**sudo systemctl daemon-reload** : permet d'appliquer les différents modifications apporter au fichier Unit

**sudo systemctl start prometheus.service** : permet de démarrer le fichier de systemd préconfigurer pour prometheus.

**sudo systemctl enable prometheus.service** : active le fichier de démarrage  
**sudo systemctl status prometheus.service -l** : vérifie si tout les configurations a pu marcher

```
jilmonde-admin@vm01:/etc/prometheus
total 28
drwxr-xr-x 2 prometheus prometheus 4096 Dec 17 2021 console_libraries
drwxr-xr-x 2 prometheus prometheus 4096 Dec 17 2021 consoles
-rw-r--r-- 1 root root 171 Jun 28 18:09 dead-mans-snitch-rule.yml
-rw-r--r-- 1 prometheus prometheus 1757 Jul 5 12:22 prometheus.yml
-rw-r--r-- 1 root root 313 Jul 9 00:00 ruleApch.yml
drwxr-xr-x 2 root root 4096 Jul 3 15:40 rules
-rw-r--r-- 1 root root 263 Jul 4 16:02 rule.yml
jilmonde-admin@vm01:/etc/prometheus$ sudo systemctl status prometheus.service
● prometheus.service - Prometheus
   Loaded: loaded (/etc/systemd/system/prometheus.service; enabled; vendor preset: enabled)
     Active: active (running) since Sun 2023-07-09 00:22:50 WAT; 1 week 2 days ago
       Main PID: 1375171 (prometheus)
         Tasks: 16 (limit: 72256)
        Memory: 68.4M
          CPU: 30min 7.279s
        CGroup: /system.slice/prometheus.service
                └─1375171 /usr/local/bin/prometheus --config.file=/etc/prometheus/prometheus.yml --storage.tsdb.path=/data

Jul 18 00:00:02 vm01 prometheus[1375171]: ts=2023-07-17T23:00:02.400Z caller=compact.go:518 level=info component=tsdb msg="compact: compacting database"
Jul 18 00:00:02 vm01 prometheus[1375171]: ts=2023-07-17T23:00:02.416Z caller=head.go:812 level=info component=tsdb msg="compact: compacted database"
Jul 18 00:00:02 vm01 prometheus[1375171]: ts=2023-07-17T23:00:02.432Z caller=checkpoint.go:98 level=info component=tsdb msg="compact: checkpointed database"
Jul 18 00:00:02 vm01 prometheus[1375171]: ts=2023-07-17T23:00:02.610Z caller=head.go:981 level=info component=tsdb msg="compact: compacted database"
Jul 18 00:00:02 vm01 prometheus[1375171]: ts=2023-07-17T23:00:02.770Z caller=compact.go:459 level=info component=tsdb msg="compact: compacted database"
Jul 18 00:00:02 vm01 prometheus[1375171]: ts=2023-07-17T23:00:02.776Z caller=db.go:1279 level=info component=tsdb msg="compact: compacted database"
Jul 18 00:00:02 vm01 prometheus[1375171]: ts=2023-07-17T23:00:02.778Z caller=db.go:1279 level=info component=tsdb msg="compact: compacted database"
Jul 18 00:00:02 vm01 prometheus[1375171]: ts=2023-07-17T23:00:02.781Z caller=db.go:1279 level=info component=tsdb msg="compact: compacted database"
Jul 18 02:00:02 vm01 prometheus[1375171]: ts=2023-07-18T01:00:02.371Z caller=compact.go:518 level=info component=tsdb msg="compact: compacted database"
Jul 18 02:00:02 vm01 prometheus[1375171]: ts=2023-07-18T01:00:02.378Z caller=head.go:812 level=info component=tsdb msg="compact: compacted database"
lines 1-20/20 (END)
```

Avec l'état « **running** » nous savons que notre fichier systemd a été bien configuré et fonctionne correctement.

Prometheus fonctionne sur le port 9090 par défaut

Avec **sudo ufw allow 9090** : on alloue le port 9090 à Prometheus.

Ensuite nous pouvons accéder à Prometheus via le navigateur en tapant l'adresse IP et le port qui dans notre cas est 161.97.162.94

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

- Address Bar:** Accès VPS Jilmonde - judithhour | Un guide pour surveiller le serveur | Systemd: Managing Linux Services | Prometheus Time Series Collector
- Title Bar:** Prometheus Time Series Collector
- Toolbar:** Accès VPS Jilmonde - judithhour | Un guide pour surveiller le serveur | Systemd: Managing Linux Services | Prometheus Time Series Collector | +
- Search Bar:** Expression (press Shift+Enter for newlines) | Execute
- Graph Panel:** Table | Graph | Evaluation time | No data queried yet | Remove Panel
- Bottom Taskbar:** Windows Start button | Taper ici pour rechercher | Various pinned application icons (File Explorer, Edge, Google Chrome, FileZilla, etc.) | Weather: 28°C | Date: 18/07/2023 | Time: 03:40

Nous pouvons maintenant afficher prometheus dans le navigateur  
Pour voir les différentes cibles de prometheus nous irons sur l'onglets « **statut** » et choisir « **targets** »

Sur la page des targets nous aurons les différentes cibles que surveille prometheus ainsi que prometheus lui-même. Lorsqu'on installe prometheus par défaut c'est lui seul qui s'affiche sur la page targets.

Notons que dans notre cas nous avons déjà ajouter d'autre cible, ne prenez pas en compte cela.

The screenshot shows a Windows desktop environment. At the top, there is a taskbar with various pinned icons. Below the taskbar, a browser window is open to the Prometheus Targets page. The page displays two tables of data. The first table, titled 'prometheus (1/1 up)', shows one endpoint: http://localhost:9100/metrics, which is UP, with labels instance="localhost:9100" and job="node\_exporter". The second table shows another endpoint: http://localhost:9090/metrics, which is UP, with labels instance="localhost:9090" and job="prometheus". Both entries have a 'Last Scrape' timestamp of approximately 10 seconds ago and a 'Scrape Duration' of about 6.789ms. The browser's address bar shows the URL as https://localhost:9090/targets. The overall interface is clean and typical of a Windows operating system.

### Interprétons cette image :

**Endpoint** : nom du job-name

**State** : statut (active « **up** » innactive « **down** »)

**Lastscrape** : indique l'heure à laquelle les métriques ont été collectées pour la dernière fois avant d'être exposées aux utilisateurs ou aux systèmes de surveillance.

**Duration** : fait référence à la période de temps pendant laquelle les données ont été collectées et agrégées.

**Error**: sur cette partie on affiche les erreurs identifier qui bloque le démarrage du métrique ou autre type d'erreur.

### Etape2 : Configuration des métriques au niveau de prometheus

- **Fonctionnement de prometheus**

Prometheus étant un système de surveillance open-source qui collecte en temps réel des métriques auprès de cibles à surveiller, comme des applications ou des infrastructures. Les données sont stockées localement sous forme de séries temporelles, accessibles via le langage de requête PromQL.

Pour collecter les métriques d'une application ou d'un service prometheus a besoin d'une « **passerelle** ». Les passerelles dans le cas de prometheus se nomme les **exporteurs**.

Il existe de nombreux types d'exportateurs pour Prometheus, chacun étant spécialement conçu pour collecter des métriques à partir de sources spécifiques.

Nous avons : **node exporter**, **MySQL Exporter**, **Redis Exporter**, **PostgreSQL Exporter**, **Apache Exporter**, **Blackbox Exporter**, etc....

Dans notre cas nous avons installer en premier lieu **node exporter**. Node exporter permet la collecte des métriques système telles que l'utilisation du CPU, de la mémoire, de l'espace disque, le réseau, etc. Il surveille les performances de l'hôte où il est installé.

## Installation de Node exporter

```
wgethttps://github.com/prometheus/node\_exporter/releases/download/v1.0.0-rc.1/node\_exporter-1.0.0-rc.1.linux-amd64.tar.gz: permet de télécharger node exporter
```

**tar -xzf node\_exporter-1.0.0-rc.1.linux-amd64.tar.gz** : dézipper le dossier node exporter téléchargé

Création d'un dossier node exporter au niveau du dossier **/usr/local/bin**

Copie du contenu du dossier node exporter dézipper vers le nouveau dossier de node exporter dans **/usr/local/bin** et donner les droits d'exécution à ce dossier.

```
jimonde-admin@vm01:/usr/local/bin
dconf          initramfs-tools  mtab          resolv.conf      vim
debcfg.conf    inputrc        mysql         resolvconfg     vsftpd.conf
debian_version inserv.conf.d  nanorc       rmt           vt100
default        iproute2       netconfig     rpc            wgetrc
deluser.conf   issue         netplan      rsyslog.conf   wpa_supplicant
depmod.d       issue.net     network      rsyslog.d      X11
dhcpc          java-11-openjdk  networkd-dispatcher sbclrc      xattr.conf
dictionaries-common java-17-openjdk  networks     security      xdg
docker         kernel        newt          selinux       zsh
dpkg          kernel-img.conf nftables.conf  sensors3.conf  zsh_command_not_found
e2scrub.conf   ldap          nginx        sensors.d     zsh-
emacs          ld.so.cache   nsswitch.conf services      shadow
environment    ld.so.conf    opt          shadow        shadow-
environment.d  ld.so.conf.d  os-release
jimonde-admin@vm01:/etc$ cd
jimonde-admin@vm01:/usr$ cd local
jimonde-admin@vm01:/usr/local$ ls
bin CHANGELOG.md etc games include lib LICENSE man n README.md sbin share src
jimonde-admin@vm01:/usr/local$ cd bin
jimonde-admin@vm01:/usr/local/bin$ ls
alertmanager  fonttools  npx      pricolpng  priplan9topng  promtool  qr          xhtml2pdf
apache_exporter ng      pip      pridiherpng  pridislch  pybidi    sqlformat
blackbox_exporter node    pip3    priforgepng  pringtonpm  _pycache_
django-admin  node_exporter pip3.10  prigreypng  prirowpng  pyfmerge  tt
django-admin.py normalizer  pisa    pripalpng   priwavepng  pyftsubset  virtualenv
dockercosmete  npm      prichunkpng  priptamtopng  prometheus  pyhanko  weasyprint
jimonde-admin@vm01:/usr/local/bin$ ls -la
total 352300
drwxr-xr-x  3 root      root          4096 Jul 14 16:24 .

```

Ensuite création du fichier systemd de node exporter afin de configurer le démarrage automatique lorsque le système d'exploitation est allumé.

```
GNU nano 6.2
[Unit]
Description=Node Exporter
Wants=network-online.target
After=network-online.target

StartLimitIntervalSec=500
StartLimitBurst=5
[Service]
User=node_exporter
Group=node_exporter
Type=simple
Restart=on-failure
RestartSec=5s
ExecStart=/usr/local/bin/node_exporter \
--collector.logind

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

**sudo systemctl daemon-reload** : redémarrer le systemd afin d'appliquer les dernières modifications

**sudo systemctl start node\_exporter** : démarrer le fichier systemd de node exporter

**sudo systemctl status node\_exporter** : vérifier si le service a bien démarrer et qu'il n'a pas d'erreur de configuration.

```
jilmonde-admin@vm01:/usr/local/bin$ sudo systemctl status node_exporter
● node_exporter.service - Node Exporter
   Loaded: loaded (/etc/systemd/system/node_exporter.service; enabled; vendor preset: enabled)
     Active: active (running) since Mon 2023-06-26 10:23:59 WAT; 3 weeks 1 day ago
       Main PID: 3684200 (node_exporter)
          Tasks: 18 (limit: 72256)
        Memory: 13.5M
           CPU: 2h 33min 49.118s
          CGroup: /system.slice/node_exporter.service
                  └─3684200 /usr/local/bin/node_exporter --collector.logind

Notice: journal has been rotated since unit was started, output may be incomplete.
jilmonde-admin@vm01:/usr/local/bin$
```

La capture d'écran ci-dessus nous indique que node exporter a été bien installer et qu'il n'a pas d'erreur de configuration d'où l'état « active » affiché.

Nous devons ajouter un nouveau travail dans le fichier prometheus.yml afin que node exporter devient une cible de prometheus.

Accéder au dossier de prometheus situé dans /etc avec la commande cd

Fait un **sudo nano prometheus.yml** pour pouvoir effectuer des modifications

```
GNU nano 6.2                               prometheus.yml
# Here it's Prometheus itself.
scrape_configs:
  # The job name is added as a label `job=<job_name>` to any timeseries scraped from this config.
  - job_name: "prometheus"
    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.
    static_configs:
      - targets: ["localhost:9090"]
  - job_name: "node_exporter"
    static_configs:
      - targets: ["localhost:9100"]
  - job_name: "apache_exporter"
    static_configs:
      - targets: ["localhost:9117"]
scrape_configs:
  - job_name: "blackbox"
    metrics_path: '/probe'
    params:
      module: [http_2xx]
    static_configs:
      - targets:
          - 161.97.162.94:8000 # Replace with the IP address and port of the Blackbox Exporter
```

Voici la configuration :

**- job\_name: 'node\_exporter'**

**static\_configs:**

**- targets: ['localhost:9100']**

**NB :** Veuillez respecter l'indentation

Après la configuration arreter et démarrer Prometheus à nouveau afin qu'il ne prenne en compte les différentes modifications.

L'exportateur node exporter que nous avons installer fonctionne sur le port 9100 donc il faut autoriser ce port au niveau de son pare feu.

Après cette configuration vous verrez que node exporter va s'afficher au niveau de vos targets dans prometheus.

The screenshot shows the Prometheus web interface with two sections: 'node\_exporter (1/1 up)' and 'prometheus (1/1 up)'. Both sections have a single table with one row. The first row contains the endpoint URL (http://localhost:9100/metrics), a green 'UP' status, and labels (instance=“localhost:9100”, job=“node\_exporter” for the first, and instance=“localhost:9090”, job=“prometheus” for the second). To the right of the table are columns for 'Last Scrape', 'Scrape Duration', and 'Error'. The 'Last Scrape' column shows the time since the last scrape (10.771s ago for node\_exporter, -2.177s ago for prometheus). The 'Scrape Duration' column shows the duration of the last scrape (22.225ms for node\_exporter, 6.789ms for prometheus). There are no errors listed. Below the tables is a search bar with placeholder text 'Taper ici pour rechercher' and a toolbar with various icons. The system tray at the bottom right shows the date (18/07/2023), time (14:06), and battery level (31°C).

**Pour l'installation des exporteurs veuillez respecter la même procédure.**

### Etape3 : Installation de Grafana

Grafana est une plateforme open-source de visualisation et d'analyse de données conçue pour créer des tableaux de bord interactifs et des graphiques pour la surveillance et l'analyse des métriques.

**sudo apt-get install -y apt-transport-https software-properties-common**: permet d'installer les packages nécessaire pour le bon fonctionnement de grafana serveur.

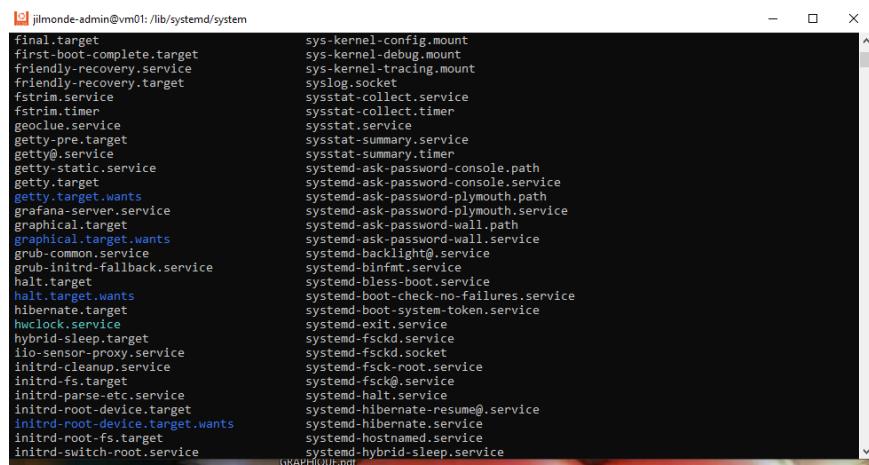
**wget -q -O - https://packages.grafana.com/gpg.key | sudo apt-key add -** : ajout de la clé GPG

**echo "deb https://packages.grafana.com/oss/deb stable main" | sudo tee -a /etc/apt/sources.list.d/grafana.list** : permet d'ajouter cette version de grafana à la liste des versions stables.

**Sudo apt-get update** : permet la mise à jour du système et des packages installer précédemment.

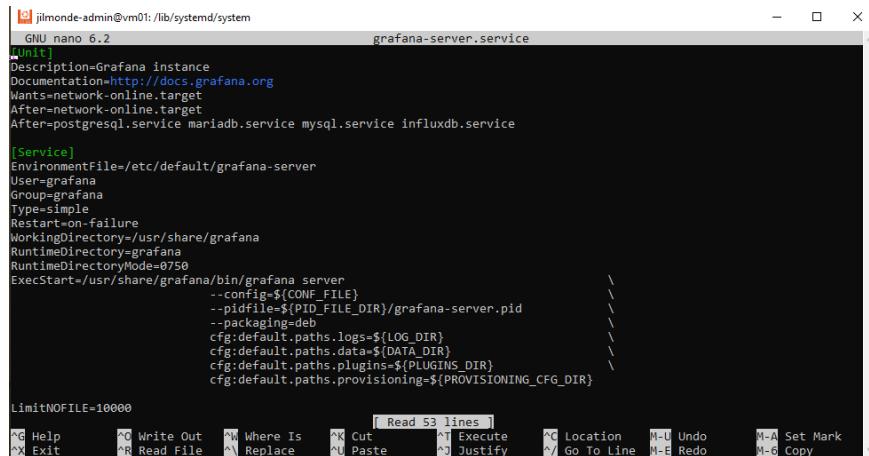
**sudo apt-get -y install grafana** : permet d'installer grafana depuis le dépôt APT.

**sudo systemctl enable grafana-server** : permet d'activer le fichier de grafana qui se situe dans /lib/systemd/system



```
jimonde-admin@vm01: /lib/systemd/system
final.target
first-boot-complete.target
friendly-recovery.service
friendly-recovery.target
fstrim.service
fstrim.timer
geoclue.service
getty-pre.target
getty@.service
getty-static.service
getty.target
getty_target.wants
grafana-server.service
graphical.target
graphical_target.wants
grub-common.service
grub-initrd-fallback.service
halt.target
halt_target.wants
hibernate.target
hwclock.service
hybrid-sleep.target
iio-sensor-proxy.service
initrd-cleanup.service
initrd-fs.target
initrd-parse-etc.service
initrd-root-device.target
initrd-root-device_target.wants
initrd-root-fs.target
initrd-switch-root.service
sys-kernel-config.mount
sys-kernel-debug.mount
sys-kernel-tracing.mount
syslog.socket
sysstat-collect.service
sysstat-collect.timer
sysstat.service
sysstat-summary.service
sysstat-summary.timer
systemd-ask-password-console.path
systemd-ask-password-console.service
systemd-ask-password-plymouth.path
systemd-ask-password-plymouth.service
systemd-ask-password-wall.path
systemd-ask-password-wall.service
systemd-backlight@.service
systemd-binfmt.service
systemd-bless-boot.service
systemd-boot-check-no-failures.service
systemd-boot-system-token.service
systemd-exit.service
systemd-fsckd.service
systemd-fsckd.socket
systemd-fsck-root.service
systemd-fsck@.service
systemd-halt.service
systemd-hibernate-resume@.service
systemd-hibernate.service
systemd-hostnamed.service
systemd-hybrid-sleep.service
```

Par défaut le fichier systemd d'un service se crée dans **lib/systemd/system** lorsqu'on utilise apt.



```
jimonde-admin@vm01: /lib/systemd/system
[Unit]
Description=Grafana instance
Documentation=http://docs.grafana.org
Wants=network-online.target
After=network-online.target
After=postgresql.service mariadb.service mysql.service influxdb.service

[Service]
EnvironmentFile=/etc/default/grafana-server
User=grafana
Group=grafana
Type=simple
Restart=on-failure
WorkingDirectory=/usr/share/grafana
RuntimeDirectory=grafana
RuntimeDirectoryMode=0750
ExecStart=/usr/share/grafana/bin/grafana server
    --config ${CONF_FILE}
    --pidfile ${PID_FILE_DIR}/grafana-server.pid
    --packaging=deb
    cfg:default.paths.logs=${LOG_DIR}
    cfg:default.paths.data=${DATA_DIR}
    cfg:default.paths.plugins=${PLUGINS_DIR}
    cfg:default.paths.provisioning=${PROVISIONING_CFG_DIR}

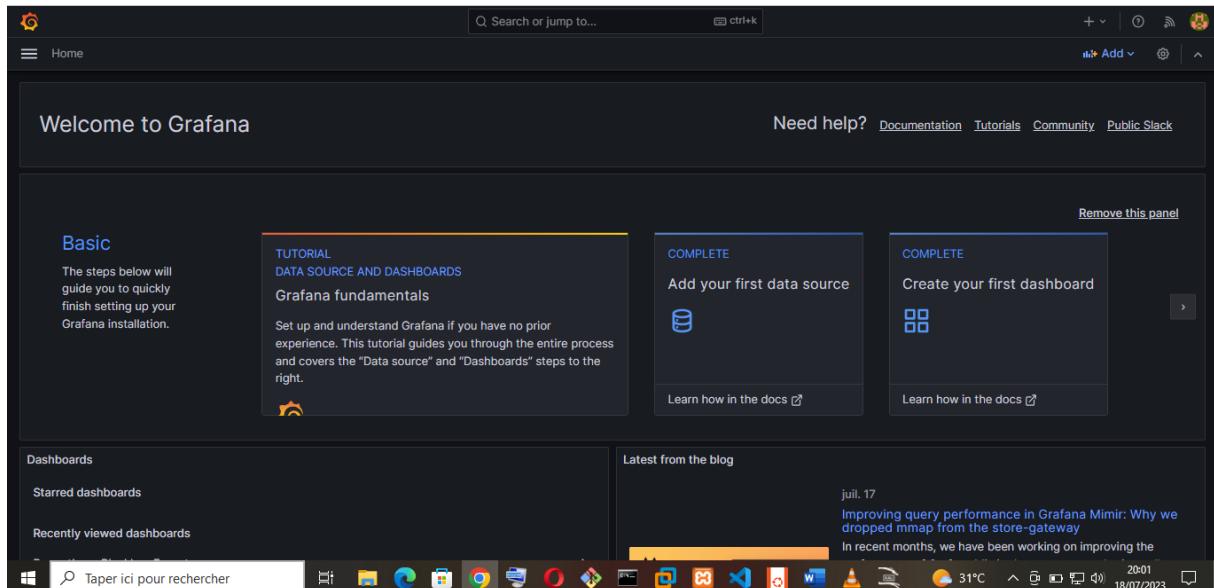
LimitNOFILE=10000
[Read 53 lines]
```

**sudo systemctl start grafana-server** : permet de démarrer le fichier systemd de grafana qui est **grafana-server.service**.

**sudo systemctl status grafana-server**

```
jilmonde-admin@vm01: /lib/systemd/system
rc-local.service          wpa_supplicant@.service
rc.service                 wpa_supplicant-wired@.service
rcS.service                x11-common.service
[jilmonde-admin@vm01: /lib/systemd/system]$ sudo nano grafana-service
[jilmonde-admin@vm01: /lib/systemd/system]$ sudo nano grafana-server
[jilmonde-admin@vm01: /lib/systemd/system]$ sudo nano grafana-server.service
[jilmonde-admin@vm01: /lib/systemd/system]$ sudo systemctl status grafana-server
● grafana-server.service - Grafana instance
   Loaded: loaded (/lib/systemd/system/grafana-server.service; enabled; vendor preset: enabled)
     Active: active (running) since Mon 2023-06-26 10:15:50 WAT; 3 weeks 1 day ago
       Docs: http://docs.grafana.org
   Main PID: 3683523 (grafana)
      Tasks: 18 (limit: 72256)
        Memory: 79.8M
         CPU: 1h 30.953s
        CGroup: /system.slice/grafana-server.service
                  └─3683523 /usr/share/grafana/bin/grafana server --config=/etc/grafana/grafana.ini --pidfile=/run/grafana/g
Jul 18 17:45:53 vm01 grafana[3683523]: logger=cleanup t=2023-07-18T17:45:53.7264600+01:00 level=info msg="Completed cleanup"
Jul 18 17:45:53 vm01 grafana[3683523]: logger=grafana.update.checker t=2023-07-18T17:45:53.873552861+01:00 level=info msg="Completed update check"
Jul 18 17:45:53 vm01 grafana[3683523]: logger=plugins.update.checker t=2023-07-18T17:45:54.0315773+01:00 level=info msg="Completed plugin update check"
Jul 18 17:55:53 vm01 grafana[3683523]: logger=cleanup t=2023-07-18T17:55:53.72799323+01:00 level=info msg="Completed cleanup"
Jul 18 17:55:53 vm01 grafana[3683523]: logger=grafana.update.checker t=2023-07-18T17:55:53.890252729+01:00 level=info msg="Completed update check"
Jul 18 17:55:54 vm01 grafana[3683523]: logger=plugins.update.checker t=2023-07-18T17:55:54.004437446+01:00 level=info msg="Completed plugin update check"
Jul 18 18:05:53 vm01 grafana[3683523]: logger=cleanup t=2023-07-18T18:05:53.72727886+01:00 level=info msg="Completed cleanup"
Jul 18 18:05:54 vm01 grafana[3683523]: logger=grafana.update.checker t=2023-07-18T18:05:53.891476487+01:00 level=info msg="Completed update check"
Jul 18 18:10:35 vm01 grafana[3683523]: logger=context userId=0 orgId=0 uname=t=2023-07-18T18:10:35.588319457+01:00 lev
lines 1-21/21 (END)
```

Grafana utilise par défaut le port 3000 donc il faut autoriser ce port au niveau de son pare feu cela permettra d'afficher grafana dans le navigateur en utilisant l'adresse IP et le port.



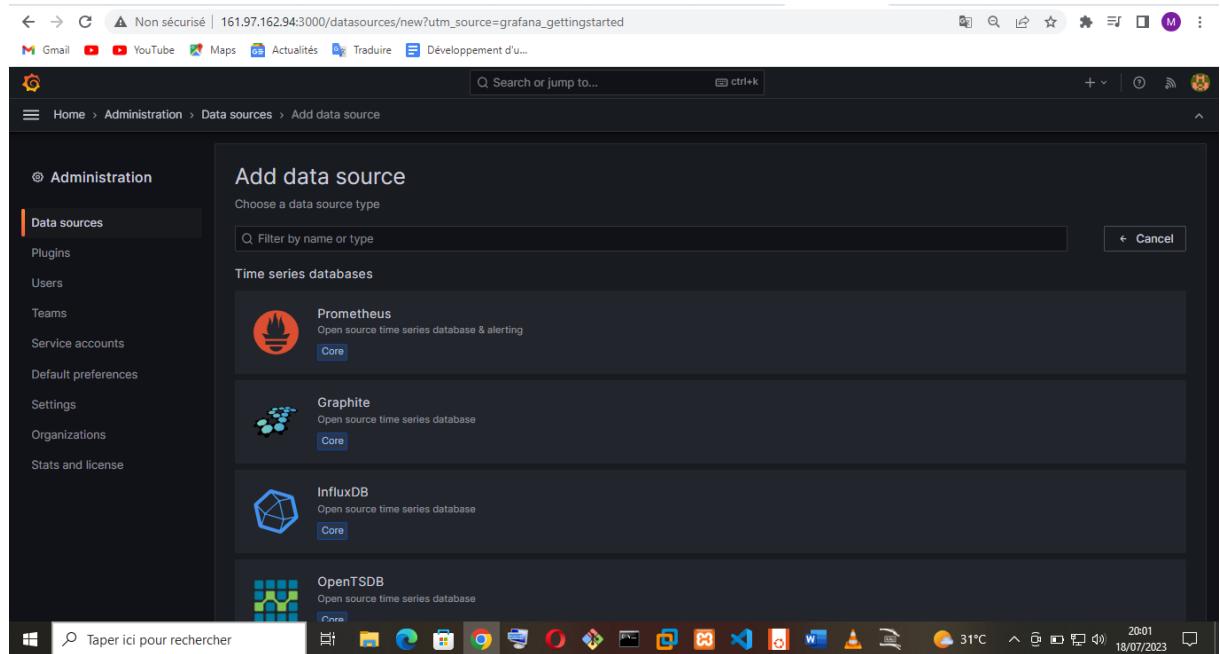
**Le mot de passe par défaut est admin et le nom d'utilisateur aussi** ; après vous êtes identifier avec les accès par défaut grafana vous demandera de mettre un nouveau mot de passe qui sera propre à votre entreprise.

Dans le cas du VPS de jilmonde le mot de passe est : jilmonde-admin

## Etape4 : intégration de prometheus à grafana

Sur Grafana cliquez sur le signe de réglage sur la gauche. Cliquez ensuite sur « **Sources de données** ».

Cliquez sur Ajouter une source de données et sélectionnez Prometheus.



Définissez l'URL comme IP du serveur Prometheus (161.97.162.94) avec le port 9090 ou localhost : 9090 si Prometheus est installer sur le même serveur que grafana.

Cliquez sur Enregistrer et tester. Vous recevrez un message de réussite de l'ajout de la source de données. (Une notification de couleur verte)

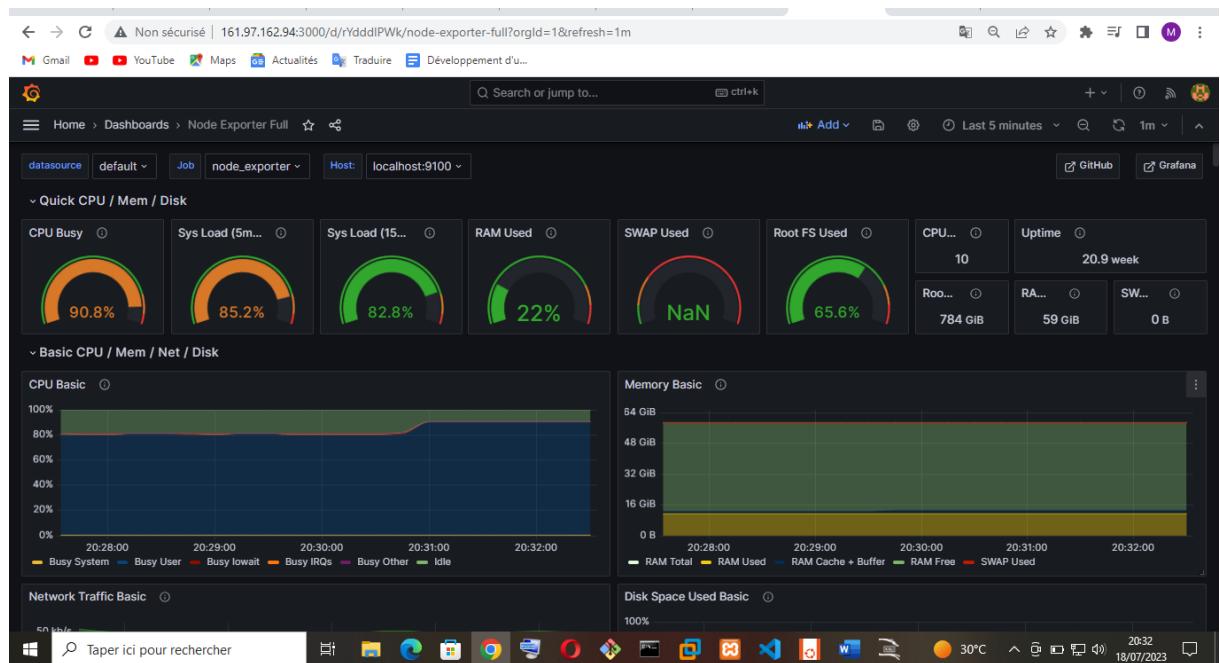
Grafana est connecté avec succès à Prometheus. Il est temps de créer un tableau de bord. Pour vous faciliter la tâche, j'utiliserai l'existant [tableau de bord de l'exportateur de nœuds](#), qui est présent sur le site officiel de Grafana pour surveiller les métriques du serveur Linux.

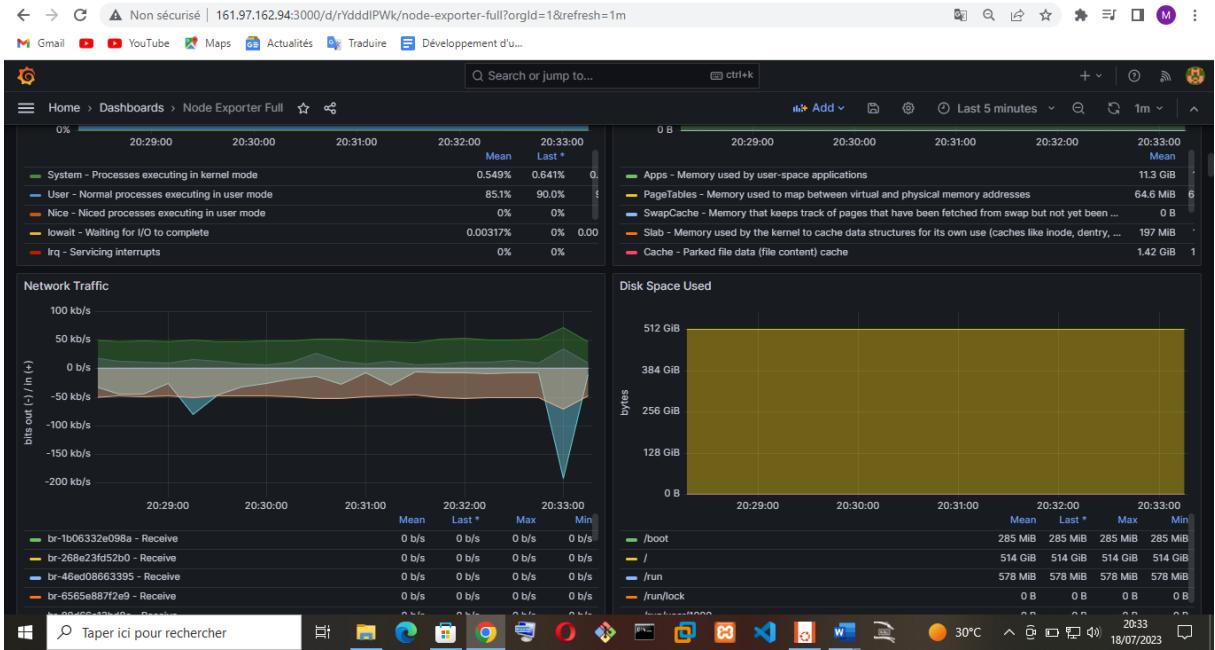
Accédez au tableau de bord d'accueil Grafana, cliquez sur + signe et cliquez sur Importer.

Dans Importer via grafana.com, mettez l'ID du tableau de bord **1860** et cliquez sur Charger.

Sélectionnez ensuite la source de données Prometheus et cliquez sur Importer.

Le tableau de bord complet de l'exportateur de nœuds est importé. Nous pouvons voir toutes les mesures telles que la charge du système, la RAM utilisée, l'occupation du processeur, etc. Qui sont surveillées avec succès sur Grafana.





## Sniffing :

### Installation et configuration de ntopng

**ntopng** est un outil de surveillance du trafic réseau qui offre une visibilité en temps réel sur le trafic réseau et l'utilisation de la bande passante.

### Ntopng :Network Top Next Generation

#### Mise en place de ntopng sur le VPS

#### Connexion au serveur de Jilmonde

```

jilmonde-admin@vm01:~$ sudo ssh 161.97.162.94@jilmonde-admin
ssh: Could not resolve hostname jilmonde-admin: Name or service not known
jilmonde-admin@vm01:~$ sudo ssh 161.97.162.94@jilmonde-admin
ssh: Could not resolve hostname jilmonde-admin: Name or service not known
jilmonde-admin@vm01:~$ sudo systemctl start ssh
Failed to start ssh.service: Unit ssh.service not found.
jilmonde-admin@vm01:~$ sudo ssh jilmonde-admin@161.97.162.94
jilmonde-admin@161.97.162.94's password:
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-70-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

 * Introducing Expanded Security Maintenance for Applications.
   Receive updates to over 25,000 software packages with your
   Ubuntu Pro subscription. Free for personal use.

   https://ubuntu.com/pro

Welcome!

This server is hosted by OPEN SOLUTIONS DIGITALES. If you have any questions or need help,
please don't hesitate to contact us at support@open.bj.

Last login: Tue Aug  8 15:53:50 2023 from 160.119.145.2
jilmonde-admin@vm01:~$
```

Sudo apt install ntopng : cette commande permet d'installer ntopng

```

jilmonde-admin@vm01:~$ sudo apt install ntopng
Unpacking libjs-jquery-ui (1.13.1+dfsg-1) ...
Selecting previously unselected package libjs-rickshaw.
Preparing to unpack .../1-libjs-rickshaw_1.5.1.1+dfsg-5_all.deb ...
Unpacking libjs-rickshaw (1.5.1.1+dfsg-5) ...
Selecting previously unselected package libndp14.2:amd64.
Preparing to unpack .../2-libndp14.2_4.2.2_2_amd64.deb ...
Unpacking libndp14.2:amd64 (4.2.2-2) ...
Selecting previously unselected package libnorm1:amd64.
Preparing to unpack .../3-libnorm1_1.5.9+dfsg-2_amd64.deb ...
Unpacking libnorm1:amd64 (1.5.9+dfsg-2) ...
Selecting previously unselected package libpbgm-5.3-0:amd64.
Preparing to unpack .../4-libpbgm-5.3-0_5.3.128+dfsg-2_amd64.deb ...
Unpacking libpbgm-5.3-0:amd64 (5.3.128+dfsg-2) ...
Selecting previously unselected package librdrd8:amd64.
Preparing to unpack .../5-librdrd8_1.7.2-3ubuntu6_amd64.deb ...
Unpacking librdrd8:amd64 (1.7.2-3ubuntu6) ...
Selecting previously unselected package libwireshark-data.
Preparing to unpack .../6-libwireshark-data_3.6.2-2_all.deb ...
Unpacking libwireshark-data (3.6.2-2) ...
Selecting previously unselected package libzmq5:amd64.
Preparing to unpack .../7-libzmq5_4.3.4-2_amd64.deb ...
Unpacking libzmq5:amd64 (4.3.4-2) ...
Selecting previously unselected package node-html5shiv.
Preparing to unpack .../8-node-html5shiv_3.7.3+dfsg-4_all.deb ...
Unpacking node-html5shiv (3.7.3+dfsg-4) ...
Selecting previously unselected package ntopng-data.
Preparing to unpack .../9-ntopng-data_5.2.1+dfsg1-1_all.deb ...
Unpacking ntopng-data (5.2.1+dfsg1-1) ...
Progress: [ 47%] [#########################################.....]
```

Après l'installation de ntopng nous sommes passé à la vérification des différentes interfaces réseau

```

jilmonde-admin@vm01:~$ ifconfig
inet6 fe80::250:56ff:fe4a:1087 brd ff:ff:ff:ff:ff:ff scopeid 0x20<link>
  ether 00:50:56:4a:10:87 txqueuelen 1000  (Ethernet)
    RX packets 8736948 bytes 4517734914 (4.5 GB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 5917996 bytes 4446483487 (4.4 GB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73  mtu 65536
  inet 127.0.0.1  netmask 255.0.0.0
    loop txqueuelen 1000  (Local Loopback)
      RX packets 18456801 bytes 6589179768 (6.5 GB)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 18456801 bytes 6589179768 (6.5 GB)
      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

vethaca6ab: flags=4163  mtu 1500
  ether 8a:ae:d4:a7:39:9c txqueuelen 0  (Ethernet)
    RX packets 4750 bytes 599474 (599.4 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 5778 bytes 4469999 (4.4 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

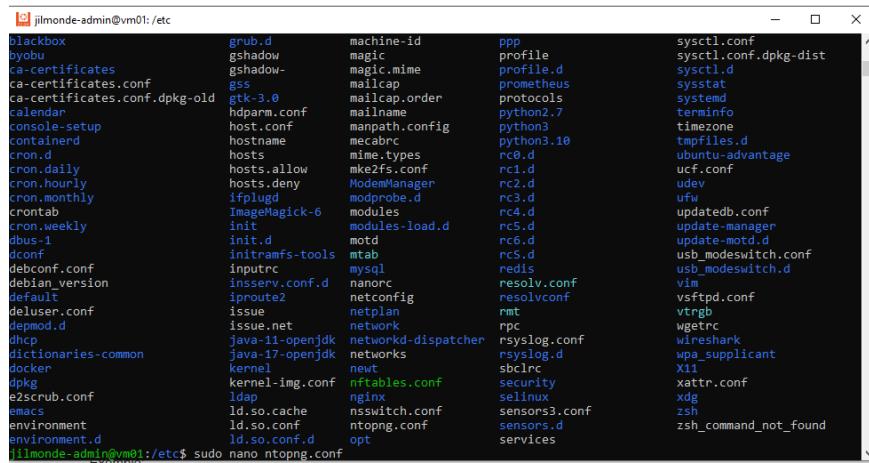
vethdabe95f: flags=4163  mtu 1500
  ether d2:c1:81:db:e9:f8 txqueuelen 0  (Ethernet)
    RX packets 21344 bytes 9662657 (9.6 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 26137 bytes 6609358 (6.6 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

jilmonde-admin@vm01:~$
```

Nous constatons que l'adresse public du serveur se situe au niveau de l'interface eth0.

Pour effectuer l'analyse du traffic il faut ajouter l'interface réseau de l'adresse IP qui est utiliser pour fournir les services.

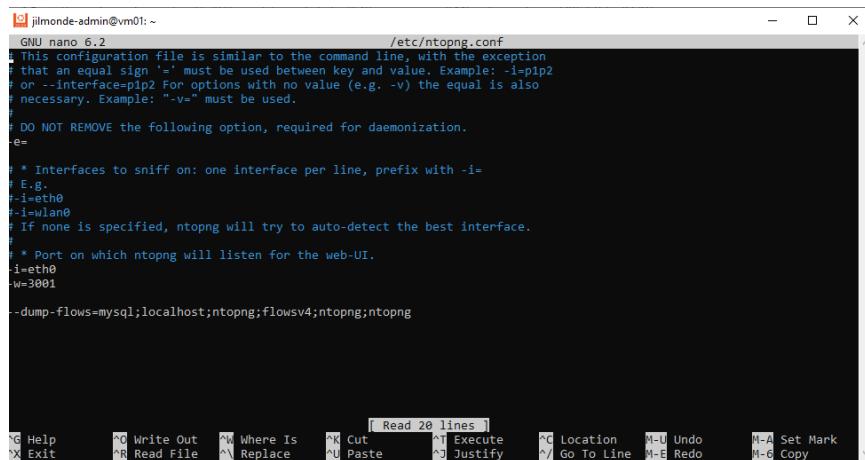
Le fichier de configuration de ntopng se situe dans le dossier /etc



```
jilmonde-admin@vm01: /etc
blackbox      grub.d      machine-id    ppp          sysctl.conf
byobu        gshadow     magic         profile      sysctl.conf.dpkg-dist
ca-certificates      gshadow-   magic.mime    profile.d    sysctl.d
ca-certificates.conf      gss       mailcap       prometheus  sysstat
ca-certificates.conf.dpkg-old      gtk-3.0  mailcap.order protocols  systemd
calendar      hdparm.conf  mailname     python2.7   terminfo
console-setup      host.conf   manpath.config  python3     timezone
containeerd      hostname   mime.types   python3.10  tmpfiles.d
cron.d        hosts      mtab         rc0.d      ubuntu-advantage
cron.daily      hosts.allow  mke2fs.conf  rc1.d      ucf.conf
cron.hourly      hosts.deny  ModemManager  rc2.d      udev
cron.monthly      iplugd     modules     rc3.d      ufw
cron.weekly      init      modules-load.d  rc4.d      updatedb.conf
crontab        init.d     motd        rc5.d      update-manager
dbus-1        initrc     mysql       rc6.d      update-motd.d
dconf        inserv.conf.d  nanorc      resolv.conf  usb_modeswitch.conf
debconf.conf      iproute2   netconfig   resolvconf  vim
debian_version      issue.net   network     rmt        vsftpd.conf
default        issue.net   network     rpc        vtrgb
deluser.conf      java-11-openjdk  networkd-dispatcher  rsyslog.conf  wgetrc
depmod.d        java-17-openjdk  networks    rsyslog.d  wireshark
dictionaries-common      kernel     neutv      sbclrc   wpa_supplicant
docker        kernel-img.conf  nftables.conf  security  X11
dpkg          ldap       nginx      selinux   xattr.conf
e2scrub.conf      ld.so.cache  nsswitch.conf  sensors3.conf  xdg
emacs        ld.so.conf   ntopng.conf  sensors.d  zsh
environment      ld.so.conf.d  opt        services   zsh_command_not_found
environment.d
jilmonde-admin@vm01: /etc$ sudo nano ntopng.conf
```

-i=eth0 : avec cette option nous demandons à ntopng de surveiller le traffic sur cette interface.

-w=3001 : permet d'ajouter à la configuration le port sur lequel ntopng doit démarrer.



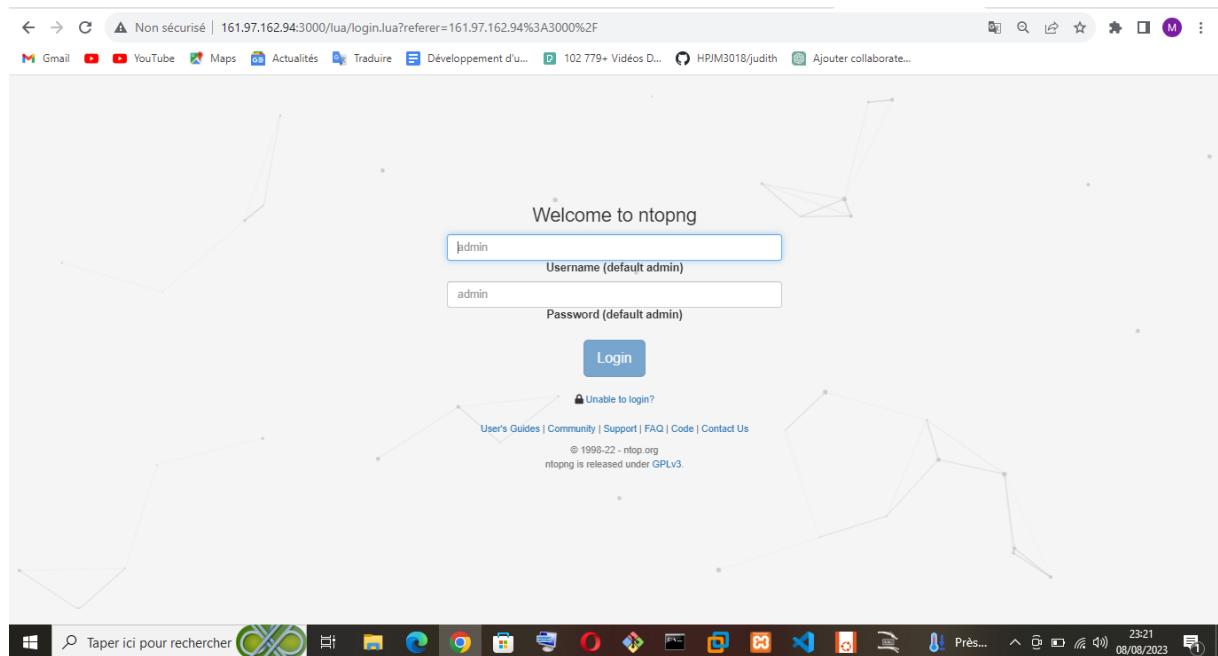
```
GNU nano 6.2                               /etc/ntopng.conf
# This configuration file is similar to the command line, with the exception
# that an equal sign '=' must be used between key and value. Example: -i=pi2
# or --Interface=pi2 For options with no value (e.g. -v) the equal is also
# necessary. Example: "-v=" must be used.
#
# DO NOT REMOVE the following option, required for daemonization.
#e=
#
# * Interfaces to sniff on: one interface per line, prefix with -i=
# E.g.
#-i=eth0
#-i=wlan0
# If none is specified, ntopng will try to auto-detect the best interface.
#
# * Port on which ntopng will listen for the web-UI.
#i=eth0
#w=3001
--dump-flows=mysql;localhost;ntopng;flows4;ntopng;ntopng
```

Nous avons ensuite activer le service ntopng avec la commande sudo systemctl enable ntopng et démarrer ntopng avec sudo systemctl start ntopng . Le service ntopng a bien démarrer sans erreur.

```
jimonde-admin@vm01:/etc$ sudo systemctl restart ntopng
jimonde-admin@vm01:/etc$ sudo systemctl status ntopng
● ntopng.service - ntopng - High-Speed Web-based Traffic Analysis and Flow Collection Tool
   Loaded: loaded (/lib/systemd/system/ntopng.service; enabled; vendor preset: enabled)
     Active: active (running) since Tue 2023-08-08 22:19:55 WAT; 10s ago
       Docs: man:ntopng(8)
                 file:/usr/share/doc/ntopng/README.Debian
                 file:/usr/share/doc/ntopng/UserGuide.pdf.gz
   Process: 377227 ExecStart=/usr/sbin/ntopng /etc/ntopng.conf (code=exited, status=0/SUCCESS)
 Main PID: 377259 (1/flow_checks)
   Tasks: 22 (limit: 72249)
    Memory: 147.8M
      CPU: 3.384s
     CGroup: /system.slice/ntopng.service
             └─377259 /usr/sbin/ntopng /etc/ntopng.conf

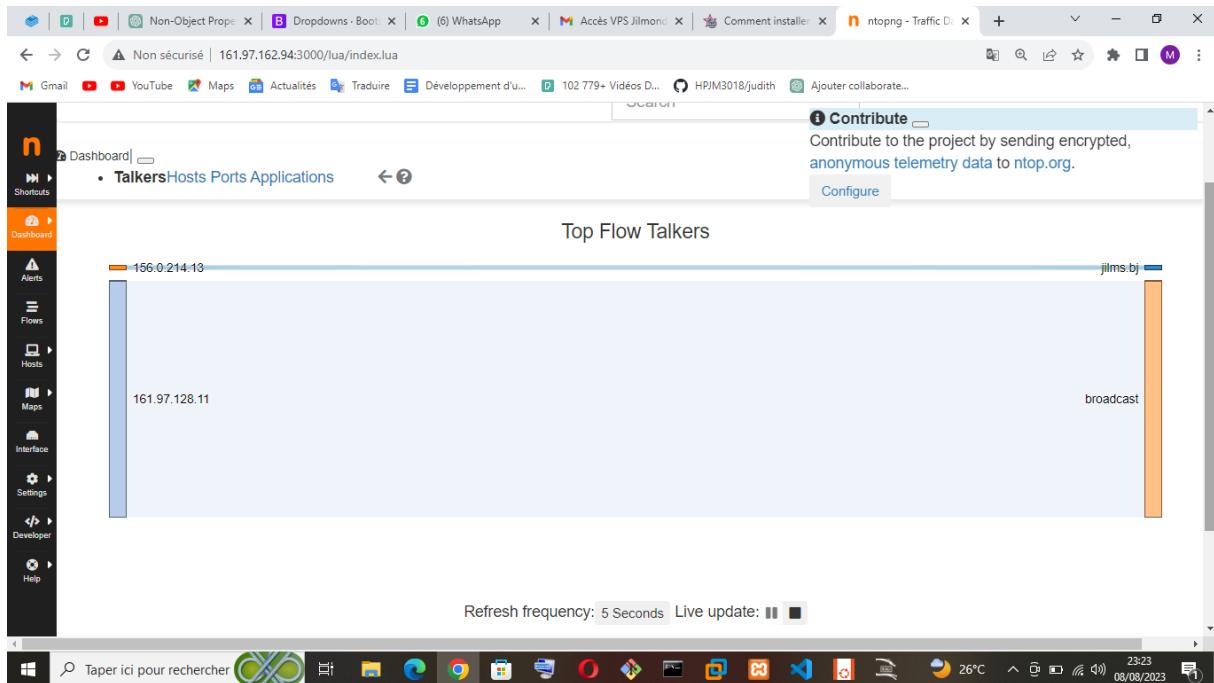
Aug 08 22:19:55 vm01 systemd[1]: Starting ntopng - High-Speed Web-based Traffic Analysis and Flow Collection Tool...
Aug 08 22:19:55 vm01 ntopng[377227]: 08/Aug/2023 22:19:55 [Ntop.cpp:3258] Added Local Network 127.0.0.0/8
Aug 08 22:19:55 vm01 ntopng[377227]: 08/Aug/2023 22:19:55 [Ntop.cpp:3258] Added Local Network fe80::/10
Aug 08 22:19:55 vm01 ntopng[377227]: 08/Aug/2023 22:19:55 [Redis.cpp:157] Successfully connected to redis 127.0.0.1:6379
Aug 08 22:19:55 vm01 ntopng[377227]: 08/Aug/2023 22:19:55 [Redis.cpp:157] Successfully connected to redis 127.0.0.1:6379
Aug 08 22:19:55 vm01 ntopng[377227]: 08/Aug/2023 22:19:55 [Ntop.cpp:2365] Parent process is exiting (this is normal)
Aug 08 22:19:55 vm01 systemd[1]: Started ntopng - High-Speed Web-based Traffic Analysis and Flow Collection Tool.
lines 1-21 (END)
```

L'état « running » nous démontre que ntopng a été bien installer et bien démarrer. Nous pouvons maintenant accéder à l'interface graphique via le navigateur en utilisant l'adresse IP et le port que nous avions configuré dans ntopng.conf



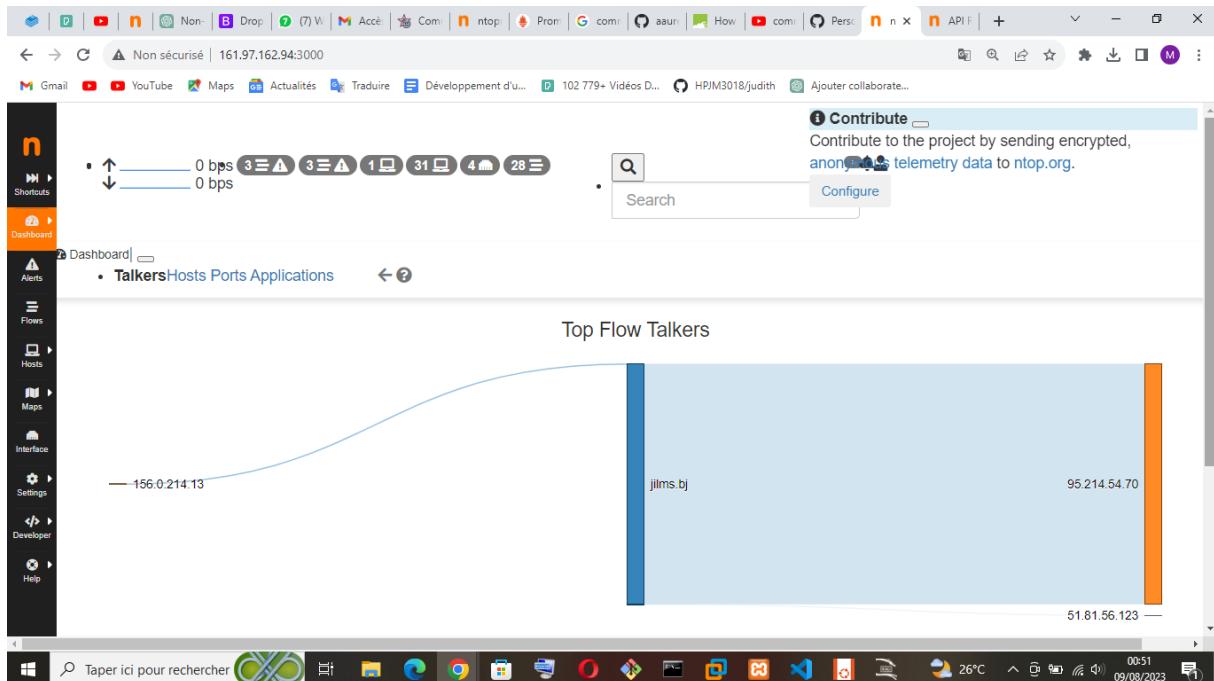
Les identifiants par défaut sont admin comme username et password.

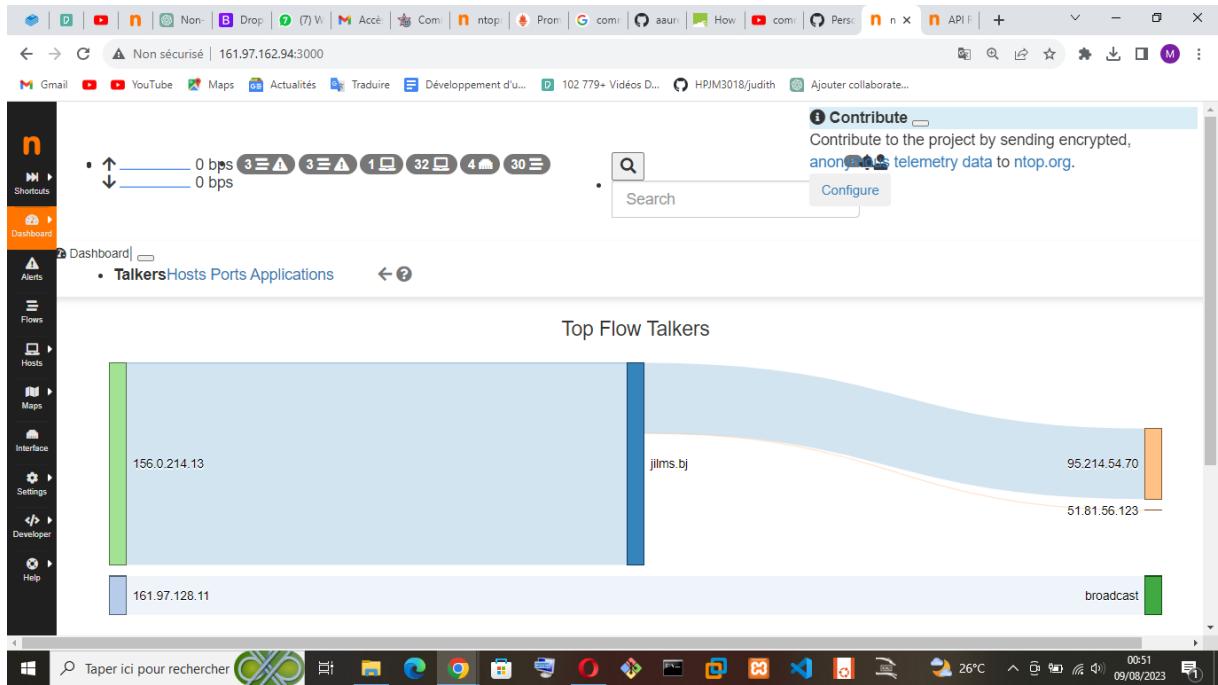
Le nouveau mot de passe est : jilmonde-admin



L'authentification à été bien effectué et nous sommes sur le dashboard de ntopng.

Voici quelques images montrant le trafic sur cette interface:





La partie de ntopng qui nous interesse afin de récupérer les informations sur le traffic est l'onglet flows.

The screenshot shows the ntopng web interface. At the top, there's a navigation bar with links to various Google services. On the left, a sidebar has sections for Shortcuts, Dashboard, Alerts, and Flow (which is currently selected). The main content area is titled "Active Flows" and contains a table with four rows of network flow data. A prominent yellow banner at the top right warns about the deprecation of MySQL support. The bottom of the screen shows a Windows taskbar with icons for various applications.

Cet onglet affiche toute les informations nécessaire (protocole utilisé, port, adresse IP source, Durée, nombre total de bytes, etc.). Pour récupérer les informations collectées par ntopng nous allons créer une base de données Mysql où nous allons stocker ses différentes informations.

## Création de la base de données ntopng

```
jimonde-admin@vm01: ~
+-----+
12 rows in set (0.001 sec)

MariaDB [(none)]> CREATE DATABASE ntopng;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near 'DATABASE ntopng' at line 1
MariaDB [(none)]> CREATE DATABASE ntopng;
Query OK, 1 row affected (0.002 sec)

MariaDB [(none)]> SHOW DATABASES;
+-----+
| Database |
+-----+
| cred_uac |
| information_schema |
| jimms_courrier |
| jimms_pack_informatique_mainenance |
| jimms_personnel |
| jimms_publication |
| jimms_tache |
| jimms_utilisateur |
| mysql |
| ntopng |
| performance_schema |
| sigfao_17072022 |
| sys |
+-----+
13 rows in set (0.001 sec)

MariaDB [(none)]>
```

## Création de l'utilisateur ntopng

```
jilmonde-admin@vm01: ~
MariaDB [(none)]> CREATE DATABASE ntopng;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near 'DATABASES ntopng' at line 1
MariaDB [(none)]> CREATE DATABASE ntopng;
Query OK, 1 row affected (0.002 sec)

MariaDB [(none)]> SHOW DATABASES;
+-----+
| Database |
+-----+
| cred_uac |
| information_schema |
| jilms_courrier |
| jilms_pack_informatique_maintenance |
| jilms_personnel |
| jilms_publication |
| jilms_tache |
| jilms_utilisateur |
| mysql |
| ntopng |
| performance_schema |
| sigfao_17072022 |
| sys |
+-----+
13 rows in set (0.001 sec)

MariaDB [(none)]> CREATE USER 'ntopng'@'%' IDENTIFIED BY 'ntopng';
Query OK, 0 rows affected (0.028 sec)

MariaDB [(none)]>
```

```
jilmonde-admin@vm01: ~
MariaDB [(none)]> SHOW DATABASES;
+-----+
| Database |
+-----+
| cred_uac |
| information_schema |
| jilms_courrier |
| jilms_pack_informatique_maintenance |
| jilms_personnel |
| jilms_publication |
| jilms_tache |
| jilms_utilisateur |
| mysql |
| ntopng |
| performance schema |
| sigfao_17072022 |
| sys |
+-----+
13 rows in set (0.001 sec)

MariaDB [(none)]> CREATE USER 'ntopng'@'%' IDENTIFIED BY 'ntopng';
Query OK, 0 rows affected (0.028 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON ntopng.* TO 'ntopng'@'%';
Query OK, 0 rows affected (0.017 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.005 sec)

MariaDB [(none)]>
```

## Création de la table flows

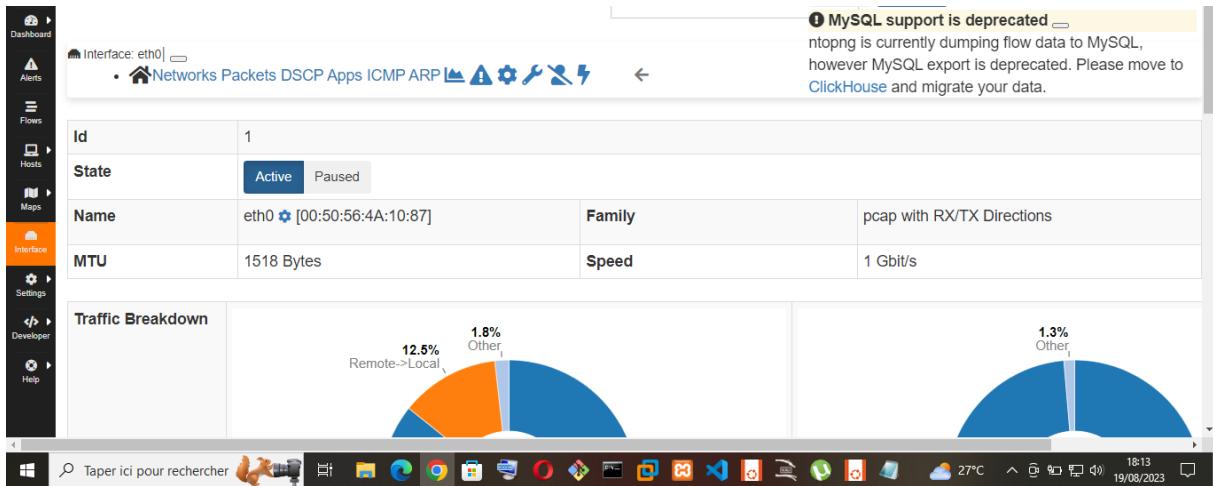
```
jilmonde-admin@vm01: ~
Query OK, 0 rows affected (0.005 sec)

MariaDB [(none)]> CREATE TABLE flows (
->     id INT AUTO_INCREMENT PRIMARY KEY,
->     source_ip VARCHAR(45),
->     destination_ip VARCHAR(45),
->     source_port INT,
->     destination_port INT,
->     protocol VARCHAR(10),
->     bytes_sent BIGINT,
->     bytes_received BIGINT,
->     timestamp TIMESTAMP
-> );
ERROR 1046 (3D000): No database selected
MariaDB [(none)]> USE ntopng;
Database changed
MariaDB [ntopng]> CREATE TABLE flows (
->     id INT AUTO_INCREMENT PRIMARY KEY,
->     source_ip VARCHAR(45),
->     destination_ip VARCHAR(45),
->     source_port INT,
->     destination_port INT,
->     protocol VARCHAR(10),
->     bytes_sent BIGINT,
->     bytes_received BIGINT,
->     timestamp TIMESTAMP
-> );
Query OK, 0 rows affected (0.028 sec)

MariaDB [ntopng]>
```

```
--dump-flows=mysql;localhost;ntopng;flows;ntopng;ntopng
^G Help ^W Write Out ^M Where Is ^K Cut ^T Execute ^C Location M-U Undo M-A Set Mark
^X Exit ^R Read File ^L Replace ^U Paste ^J Justify ^Y Go To Line M-E Redo M-6 Copy
ur recherche 27°C
```

Cette ligne notifie à ntopng d'envoyer les informations collecté et affiché au niveau dans notre base de données Mysql.



Le dump à marcher avec succès donc nos informations sont envoyé à notre base de donnée MySql.

```
jilmonde-admin@vm01: ~
This server is hosted by OPEN SOLUTIONS DIGITALES. If you have any questions or need help,
please don't hesitate to contact us at support@open.bj.

Last login: Sat Aug 19 16:52:12 2023 from 156.0.214.37
jilmonde-admin@vm01:~$ sudo mysql -u root
[sudo] password for jilmonde-admin:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 314150
Server version: 10.6.12-MariaDB-0ubuntu0.22.04.1 Ubuntu 22.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

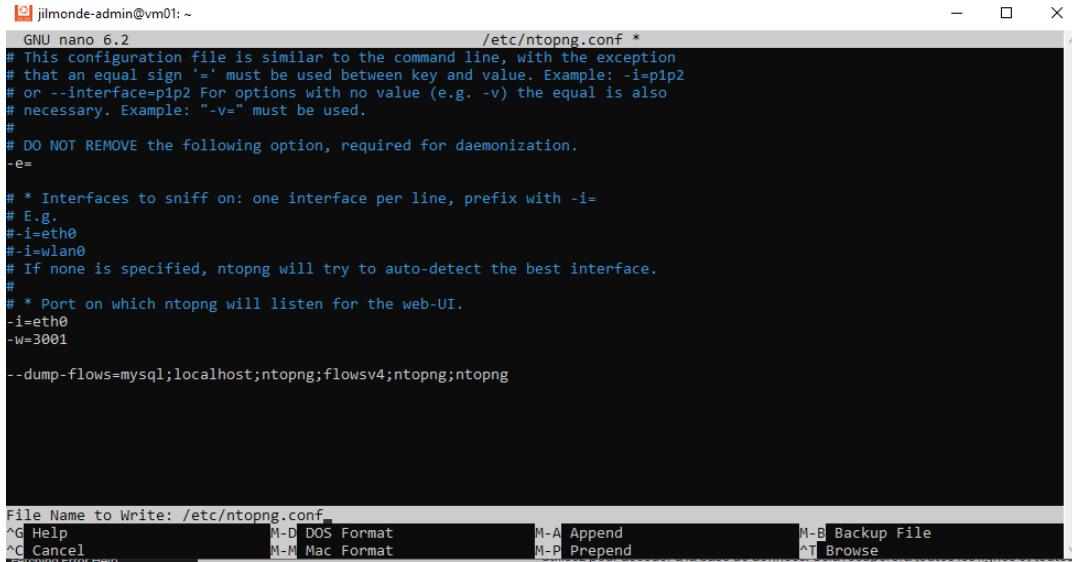
MariaDB [(none)]> USE ntopng;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [ntopng]> SHOW TABLES;
+-----+
| Tables_in_ntopng |
+-----+
| flows           |
| flowsv4         |
| flowsv6         |
+-----+
3 rows in set (0.000 sec)

MariaDB [ntopng]>
```

Deux autres tables ont été ajouter de manière automatique, c'est deux tables flowsv4 et flowsv6 contiennent le flux lorsqu'on utilise IPv4 et IPV6.

Nous devons ajouter la table flowsv4 au niveau de la configuration de ntopng

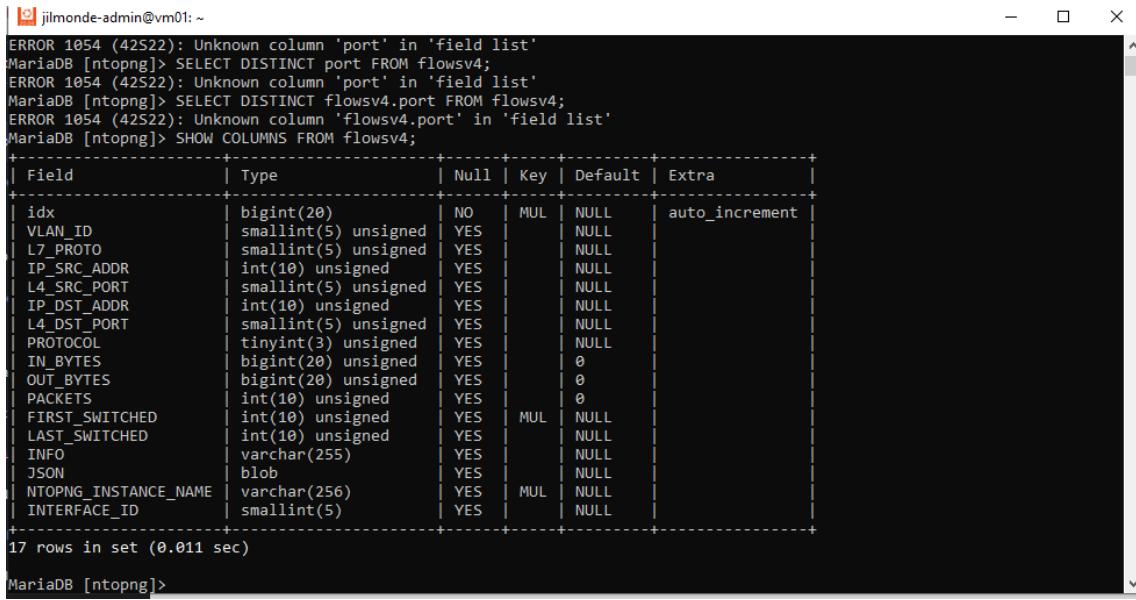


```
GNU nano 6.2                               /etc/ntopng.conf *
# This configuration file is similar to the command line, with the exception
# that an equal sign '=' must be used between key and value. Example: -i=pipl2
# or --interface=pipl2 For options with no value (e.g. -v) the equal is also
# necessary. Example: "-v=" must be used.
#
# DO NOT REMOVE the following option, required for daemonization.
-e

# * Interfaces to sniff on: one interface per line, prefix with -i=
# E.g.
#-i=eth0
#-i=wlan0
# If none is specified, ntopng will try to auto-detect the best interface.
#
# * Port on which ntopng will listen for the web-UI.
-i=eth0
-w=3001

--dump-flows=mysql;localhost;ntopng;flowsv4;ntopng;ntopng
```

Dans notre cas c'est flowsv4 qui nous concerne voyons le contenu de la table flowsv4.



```
jilmonde-admin@vm01: ~
ERROR 1054 (42S22): Unknown column 'port' in 'field list'
MariaDB [ntopng]> SELECT DISTINCT port FROM flowsv4;
ERROR 1054 (42S22): Unknown column 'port' in 'field list'
MariaDB [ntopng]> SELECT DISTINCT flowsv4.port FROM flowsv4;
ERROR 1054 (42S22): Unknown column 'flowsv4.port' in 'field list'
MariaDB [ntopng]> SHOW COLUMNS FROM flowsv4;
+-----+-----+-----+-----+-----+
| Field      | Type       | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+
| idx        | bigint(20) | NO   | MUL | NULL    | auto_increment |
| VLAN_ID    | smallint(5) | YES  |     | NULL    |                |
| L7_PROTO    | smallint(5) | YES  |     | NULL    |                |
| IP_SRC_ADDR| int(10) unsigned | YES  |     | NULL    |                |
| L4_SRC_PORT | smallint(5) | YES  |     | NULL    |                |
| IP_DST_ADDR| int(10) unsigned | YES  |     | NULL    |                |
| L4_DST_PORT | smallint(5) | YES  |     | NULL    |                |
| PROTOCOL   | tinyint(3) unsigned | YES  |     | NULL    |                |
| IN_BYTES    | bigint(20) | YES  |     | 0       |                |
| OUT_BYTES   | bigint(20) | YES  |     | 0       |                |
| PACKETS    | int(10) unsigned | YES  |     | 0       |                |
| FIRST_SWITCHED | int(10) unsigned | YES  | MUL | NULL    |                |
| LAST_SWITCHED | int(10) unsigned | YES  |     | NULL    |                |
| INFO        | varchar(255) | YES  |     | NULL    |                |
| JSON        | blob        | YES  |     | NULL    |                |
| NTOPNG_INSTANCE_NAME | varchar(256) | YES  | MUL | NULL    |                |
| INTERFACE_ID | smallint(5) | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+
17 rows in set (0.011 sec)

MariaDB [ntopng]>
```

Il y a 17 colonne dans notre table flowsv4

Chaque colonne contient les enregistrements comme au niveau de flows dans ntopng ;

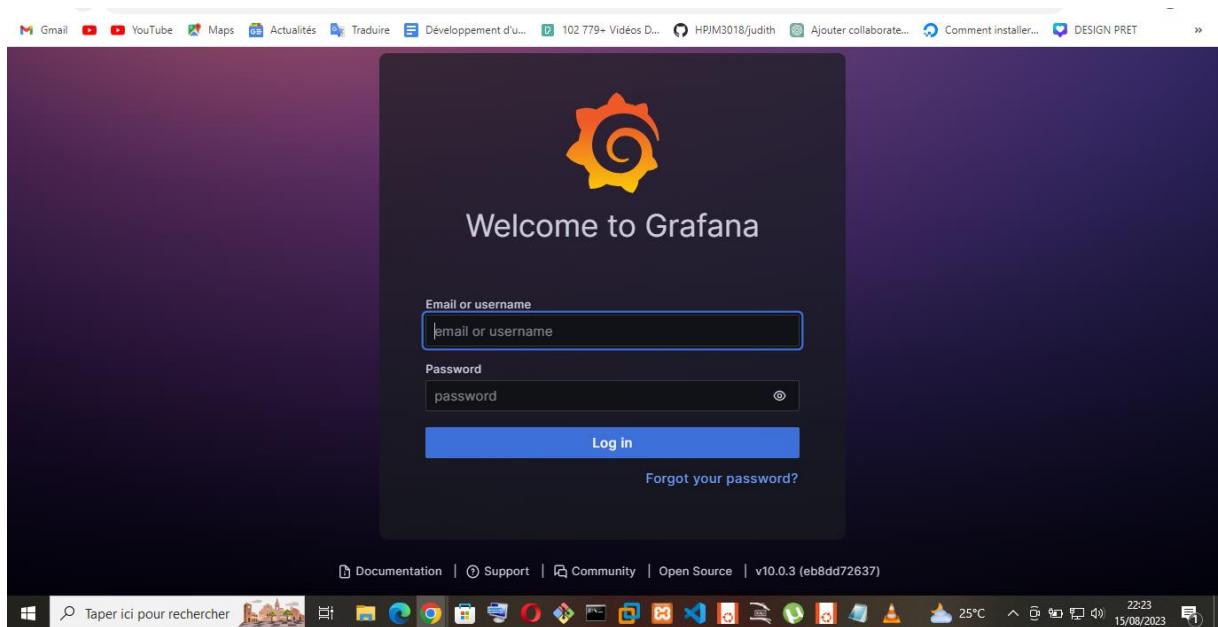
Exemple voici le contenu de L4\_DST\_PORT ( port de destination au niveau de notre serveur).

```
jilmonde-admin@vm01: ~
4432
10554
443
8086
4002
32253
22
60002
445
49200
769
22222
46159
3389
19291
5613
443
68
443
33389
54041
50107
22
1386
22
62362
443
22
13020
9527
```

## Intégration de ntopng à Grafana

Connexion à grafana avec

Username : admin et mot de passe : jilmonde-admin



L'étape suivante est d'ajouter la base de données Mysql de ntopng comme source de donnée à grafana.

The screenshot shows the Grafana interface for managing data sources. On the left, a sidebar menu is open under 'Administration' with 'Data sources' selected. In the main content area, a MySQL connection is being configured. The connection is named 'ntopng-dynamics'. The 'Alerting supported' status is shown as 'Not supported'. The 'MySQL Connection' section contains the following fields:

- Host: localhost:3306
- Database: ntopng
- User: ntopng (password masked)
- Session timezone: (default)
- Use TLS Client Auth: Off
- With CA Cert: Off
- Skip TLS Verification: Off

At the bottom right of the connection configuration, there are 'Delete' and 'Save & test' buttons.

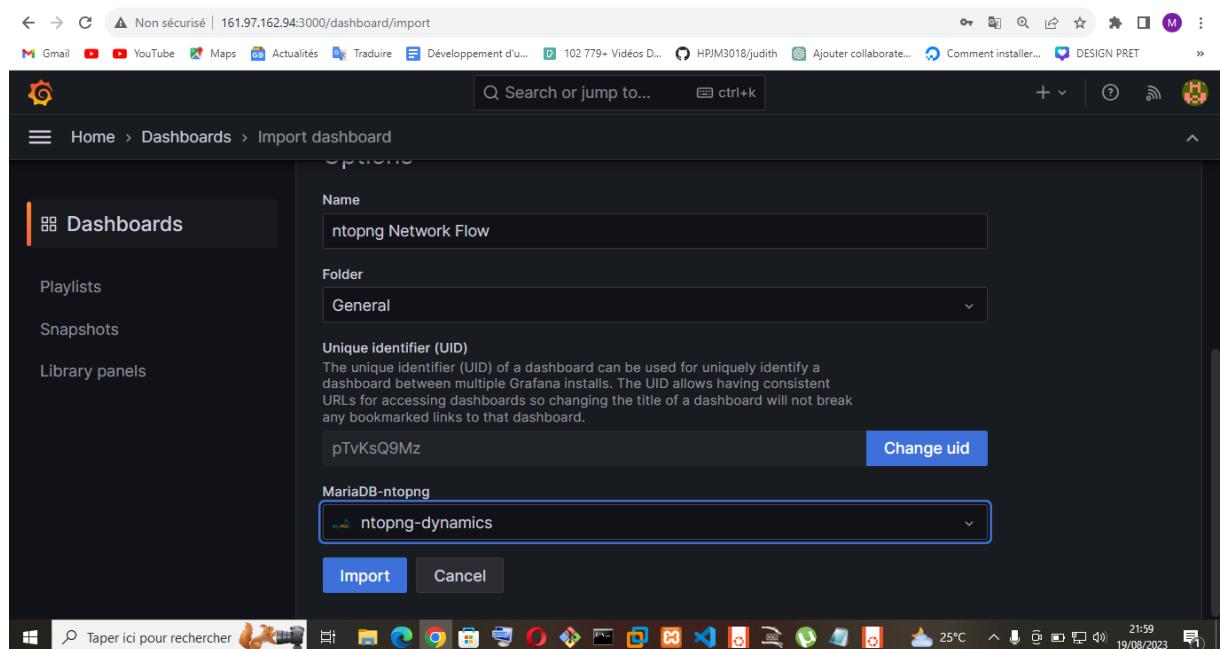
Le port d'écoute de Mysql est par défaut est 3306, au niveau de database nous avions insérer le nom de la base de donnée ntopng qui est ntopng dans notre cas. Le « user » correspond au nom d'utilisateur ayant les droits sur notre base donné et le mot de passe correspond à son mot de passe.

The screenshot shows the results of the MySQL connection test. A message box indicates 'Database Connection OK' with a green checkmark icon. The message states: 'Next, you can start to visualize data by [building a dashboard](#), or by querying data in the [Explore view](#)'. Below this message, there are 'Delete' and 'Save & test' buttons. The left sidebar shows the 'Data sources' section is still selected.

Ensuite cliquer sur « save et test » pour ajouter la base de données comme source de donnée à grafana.

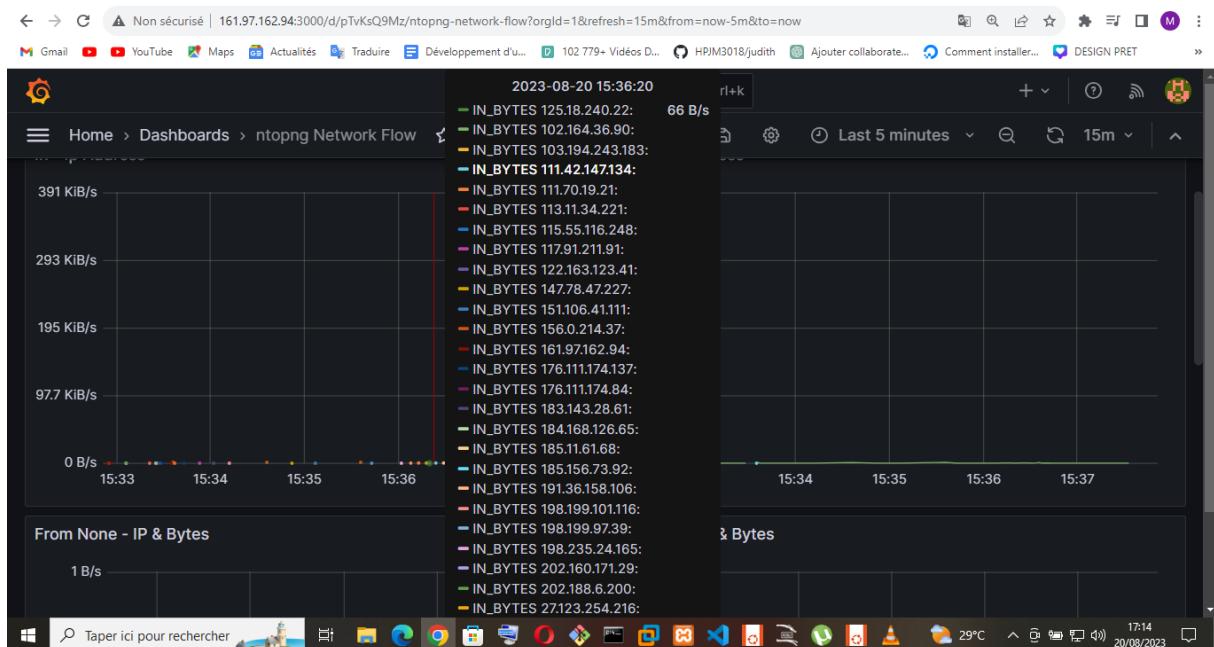
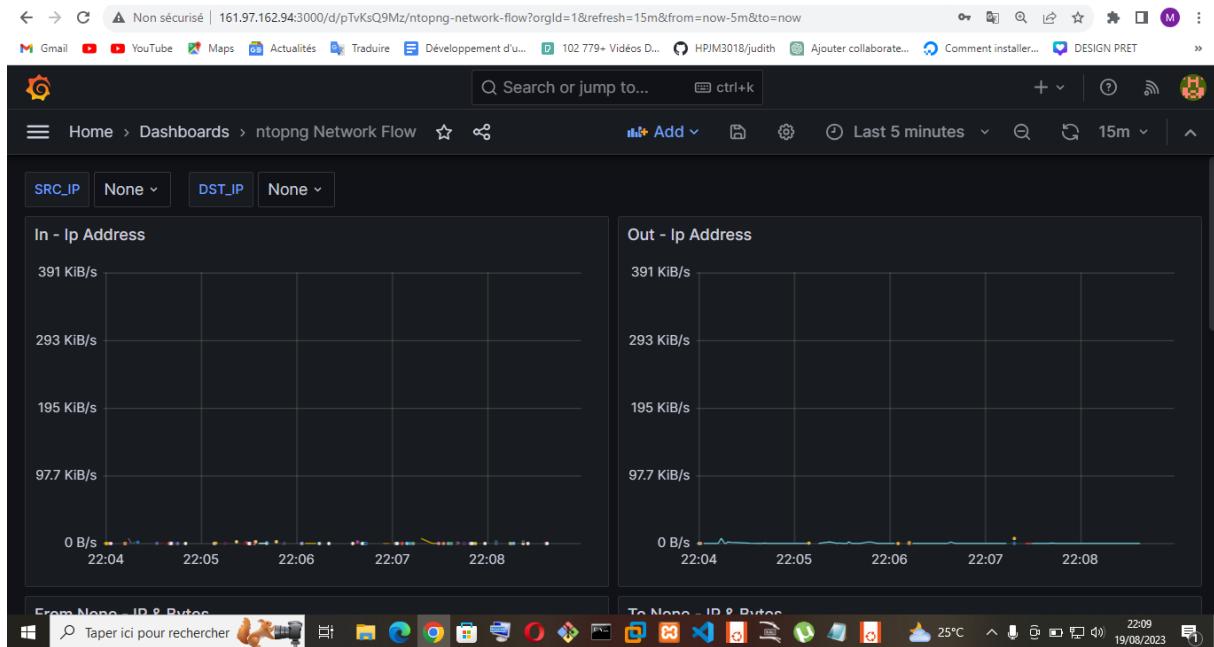
## Création du dashboard

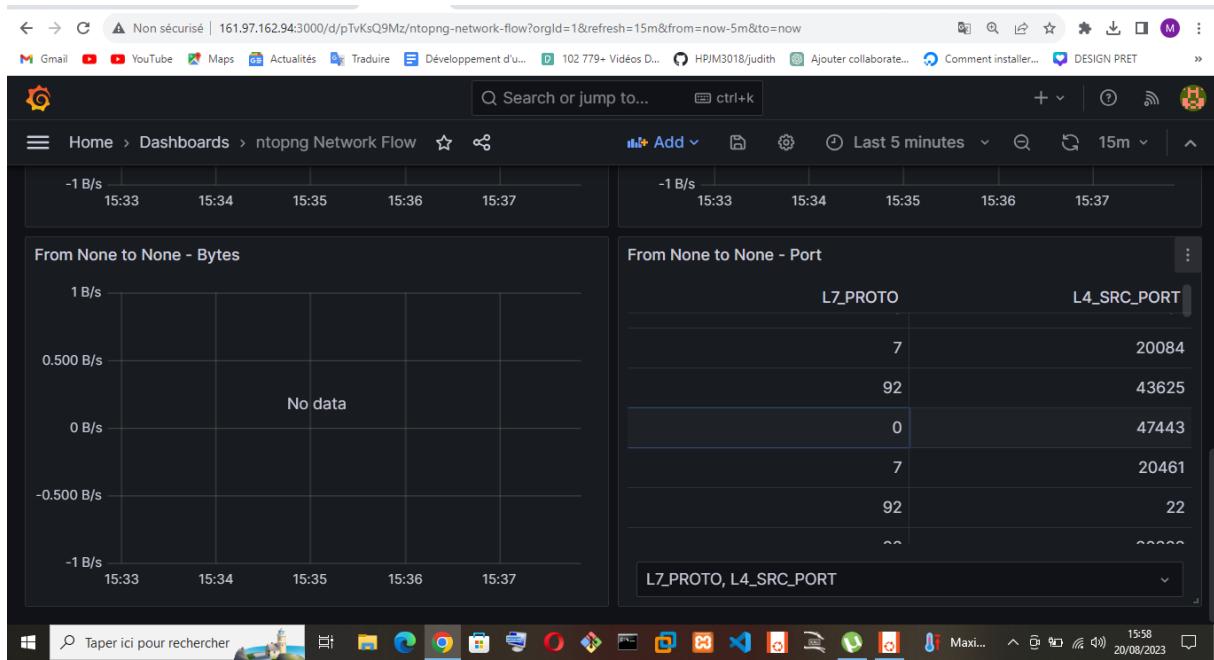
Il existe un dashboard développer par l'équipe de grafana qui répond à notre besoin donc nous allons l'importer. L'identifiant de cette base de données est : 14357.



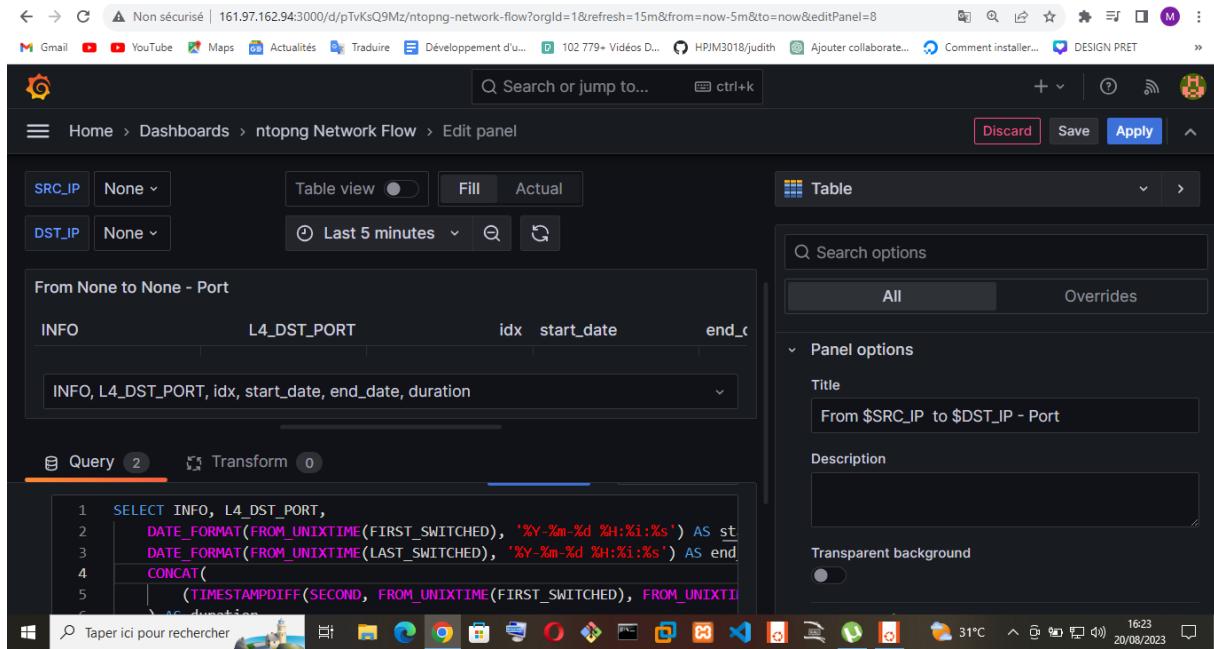
Comme source lors de l'import du Dashboard il faut spécifier la source créer précédemment.

Voici quelques images du dashboard importer





Nous pouvons utiliser des requêtes SQL pour afficher les informations rechercher au niveau de flows dans grafana.



Voici un exemple :

Une requête SQL qui permet de récupérer la durée, le port de destination ainsi que la rubrique information au niveau de ntopng. La rubrique info donne plus d'information sur l'action effectuée.

Non sécurisé | 161.97.162.94:3000/d/pTvKsQ9Mz/ntopng-network-flow?orgId=1&refresh=15m&from=now-5m&to=now

Gmail YouTube Maps Actualités Traduire Développement d'u... 102 779+ Vidéos D... HPJM3018judith Ajouter collaborat... Comment installer... DESIGN PRET

Home > Dashboards > ntopng Network Flow

From None to None - Bytes

From None to None - Port

_PORT	start_date	end_date	duration
30171	2023-08-19 17:45...	2023-08-19 17:45...	0h 0m 21s
30167	2023-08-19 17:45...	2023-08-19 17:45...	0h 0m 21s
28358	2023-08-19 17:45...	2023-08-19 17:45...	0h 0m 21s
28357	2023-08-19 17:45...	2023-08-19 17:45...	0h 0m 21s
63594	2023-08-19 17:45...	2023-08-19 17:45...	0h 0m 22s
17674	2023-08-19 17:45...	2023-08-19 17:45...	0h 0m 22s

INFO, L4\_DST\_PORT, start\_date, end\_date, duration

Taper ici pour rechercher 16:29 20/08/2023

Non sécurisé | 161.97.162.94:3000/d/pTvKsQ9Mz/ntopng-network-flow?orgId=1&refresh=15m&from=now-5m&to=now

Gmail YouTube Maps Actualités Traduire Développement d'u... 102 779+ Vidéos D... HPJM3018judith Ajouter collaborat... Comment installer... DESIGN PRET

Home > Dashboards > ntopng Network Flow

From None to None - Port

INFO	IP_SRC_ADDR	L4_DST_PORT	start_date
	1509467011	50537	2023-08-19 17:02:00
keyserver.ubuntu....	2707530334	11371	2023-08-19 17:02:00
	2707530334	123	2023-08-19 17:02:00
	1984342315	445	2023-08-19 17:02:00
keyserver.ubuntu....	2707530334	53	2023-08-19 17:02:00

INFO, IP\_SRC\_ADDR, L4\_DST\_PORT, start\_date, end\_date, duration

Taper ici pour rechercher 17:14 20/08/2023

Nous pouvons ajouter d'autre rows à notre dashboard et afficher d'autre requêtes



