



# EXPLANATORY DOCUMENTATION

Debusschere Antoine  
Tridon Quentin  
PANDRAUD Nicolas  
SCHERPEREEL Clement  
CHATEAU Nicolas

## Summary

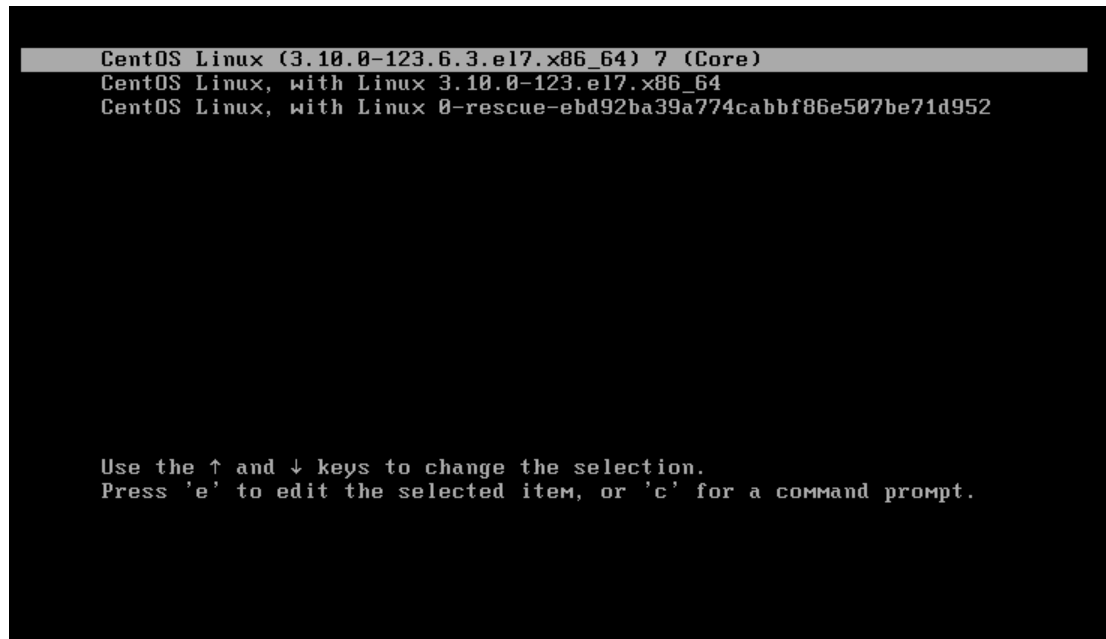
Grub-2 (BOOTHOLE) (Hack).....	4
Hacking part .....	4
In the boot grub menu select option to edit. ....	4
Select Option to edit (e). ....	4
Go to the line of Linux 16 and change ro with rw init=/sysroot/bin/sh. ....	5
Now press Control+x to start on single user mode. ....	5
Now access the system with this command. ....	5
Reset the password.....	5
Update selinux information.....	5
Exit chroot.....	5
Reboot your system .....	5
Nagios recover admin password .....	6
Portainer recover soupeladmin password.....	6
Portainer recover admin password (BRUTE) .....	8
Install Gitlab with docker.....	9
Gitlab runner.....	9
How to install NRPE AGENT (Nagios) .....	11
Client Side .....	11
Supported Distributions.....	11
Installing The Agent.....	11
Server side .....	11
Install Grafana & Prometheus.....	15
Grafana install with docker.....	15
Prometheus install with Node Explorer.....	15
Import specific Dashboard.....	19
Install Mysql Exporter.....	21
Firewall configuration .....	23
Matrice de Flux .....	23
Installation.....	24
Flow Blockage.....	24
Suppression of services and ports .....	24
Add an exception.....	24
Add a service or a port in exception.....	24
Restart FirewallD .....	24
Revert to default (for reset) .....	24

Docker restart always .....	25
NTP synchronized : yes.....	25
Update root password on mysql .....	25
Change a user password.....	26

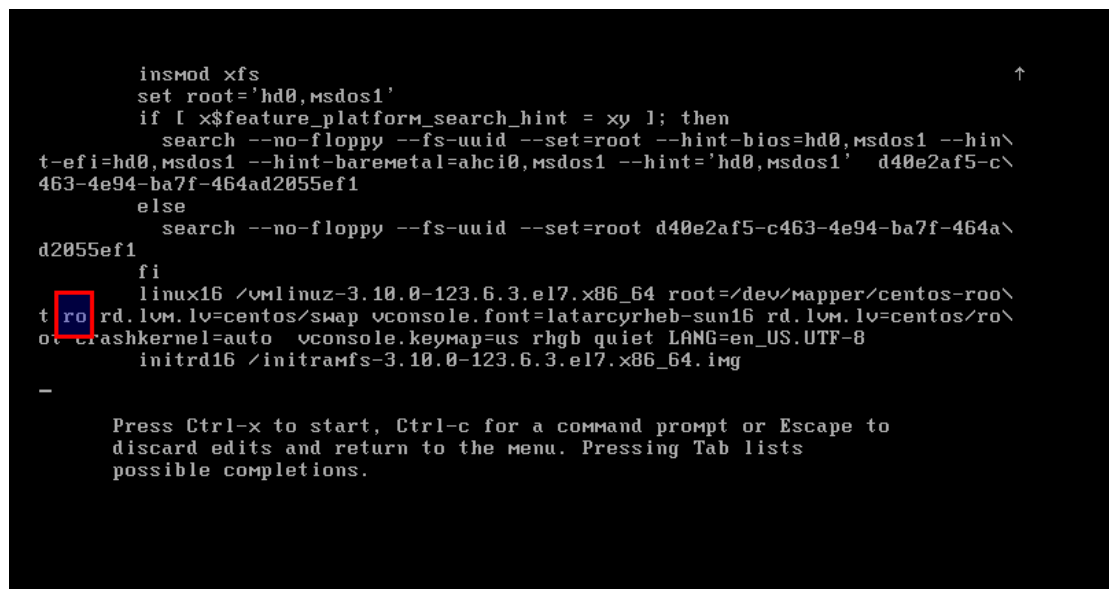
# Grub-2 (BOOTHOLE) (Hack)

## Hacking part

In the boot grub menu select option to edit.



Select Option to edit (e).



Go to the line of Linux 16 and change ro with rw init=/sysroot/bin/sh.

```
insmod xfs
set root='hd0,msdos1'
if [ x$feature_platform_search_hint = xy 1; then
    search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos1 --hin\
t-efi=hd0,msdos1 --hint-baremetal=ahci0,msdos1 --hint='hd0,msdos1' d40e2af5-c\
463-4e94-ba7f-464ad2055ef1
else
    search --no-floppy --fs-uuid --set=root d40e2af5-c463-4e94-ba7f-464a\
d2055ef1
fi
linux16 /vmlinuz-3.10.0-123.6.3.el7.x86_64 root=/dev/mapper/centos-root\
t rw init=/sysroot/bin/sh rd.lvm.lv=centos/swap vconsole.font=latarcyrheb-sun\
16 rd.lvm.lv=centos/root crashkernel=auto vconsole.keymap=us rhgb quiet LANG=\
en_US.UTF-8
initrd16 /initramfs-3.10.0-123.6.3.el7.x86_64.img

Press Ctrl-x to start, Ctrl-c for a command prompt or Escape to
discard edits and return to the menu. Pressing Tab lists
possible completions.
```

Now press Control+x to start on single user mode.

```
[ OK ] Started Show Plymouth Boot Screen.
[ OK ] Reached target Paths.
[ OK ] Reached target Basic System.
Starting File System Check on /dev/mapper/centos-root...
[ 1.062057] sd 0:0:0:0: [sda] Assuming drive cache: write through
[ 1.063004] sd 0:0:0:0: [sda] Assuming drive cache: write through
[ 1.063978] sd 0:0:0:0: [sda] Assuming drive cache: write through
systemd-fsck[394]: fsck: error 2 (No such file or directory) while executing fsck.ext2 for /dev/mapper/centos-root
[ OK ] Started File System Check on /dev/mapper/centos-root.
[ OK ] Started dracut initqueue hook.
Mounting /sysroot...
[ OK ] Mounted /sysroot.
[ OK ] Reached target Initrd Root File System.
Starting Reload Configuration from the Real Root...
[ OK ] Started Reload Configuration from the Real Root.
[ OK ] Reached target Initrd File Systems.
[ OK ] Reached target Initrd Default Target.

Generating "/run/initramfs/rdsosreport.txt"

Entering emergency mode. Exit the shell to continue.
Type "journalctl" to view system logs.
You might want to save "/run/initramfs/rdsosreport.txt" to a USB stick or /boot
after mounting them and attach it to a bug report.

:/# _
```

Now access the system with this command.

chroot /sysroot

Reset the password.

passwd root

Update selinux information

touch /.autorelabel

Exit chroot

exit

Reboot your system

Reboot

## Nagios recover admin password

```
[root@localhost ~]# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
e1ec5de2d636   nagiosxi:5.5.6 "/usr/local/bin/run..." 13 months ago Up 39 minutes 0.0.0.0:80->80/tcp               nostalgic_ganguly
ab1349a18882   mysql:5.7      "docker-entrypoint.s..." 13 months ago Up 39 minutes 0.0.0.0:3306->3306/tcp, 33060/tcp dev_db
```

You need to go in :

```
docker exec -it e1ec5de2d636 /bin/bash
```

then

```
/usr/local/nagiosxi/scripts/reset_nagiosadmin_password.php --password=newpassword
```

## Portainer recover soupeladmin password

Admin n'existe pas, sur le fichier de portainer.db nous avons trouvé l'utilisateur suivant :

```
{ "Id": 2, "Username": "souveladmin", "Password": "$2a$10$QftFvTx5yBpf06i0oqT6G.Nmc0CFB6GPiHVbCBq4MbcfxsQV", "PortainerEndpointList": true, "PortainerExtensionList": true, "PortainerMOTD": true, "PortainerRegistryInspect": true, "PortainerRegistryList": true }
```

Pour recouvrer son mot de passe

```
yum install unzip && yum install curl  
curl -sSL
```

```
https://git.bullercodeworks.com/brian/boltbrowser/releases/download/2.0/boltbrowser.linux64.zip -o bolt.zip
```

```
unzip bolt.zip
```

```
docker volume ls
```

```
docker stop portainer
```

```
docker run --rm httpd:2.4-alpine htpasswd -nbB soupeladmin 'password'
```

```
[root@localhost recover]# docker run --rm httpd:2.4-alpine htpasswd -nbB soupeladmin 'password'
Unable to find image 'httpd:2.4-alpine' locally
2.4-alpine: Pulling from library/httpd
f84cab65f19f: Pull complete
bb2b121e3f63: Pull complete
5b78361913e7: Pull complete
32af7daef0d3: Pull complete
b9d08706c705: Pull complete
Digest: sha256:8c16a28de3e8a715c613bc84868f8ccb984eca1027800f18bfc7a0fab377f475
Status: Downloaded newer image for httpd:2.4-alpine
souveladmin:$2y$05$DFHzEnhTkiilUJFjE8uvXOwMvDr8n6o8kMqgRfNr/2JwLqFI4MrIO
```

Copy the hash and

```
./boltbrowser.linux64 /var/lib/docker/volumes/portainer_data/_data/portainer.db
```

```

boltbrowser: /var/lib/docker/volumes/portainer_data/_data/portainer.db
=====
+ dockerhub
+ endpoint_groups
+ endpoints
+ extension
+ registries
+ resource_control
+ roles
+ schedules
+ settings
+ stacks
+ tags
+ team_membership
+ teams
+ templates
+ tunnel_server
+ users
2: {"Id":2,"Username":"soupeLadmin","Password":"$2a$10$0tfvTx5yBpf0610oqT6G.Nmc8CFB6GP1HVbCBq4MbcfxsQVeV9HS",
3: {"Id":3,"Username":"no_rights","Password":"$2a$10$pv11HuFMH5dt4/uMu0J1T.wLdBNiLOH8ZMMC
+ version
+ webhooks

Path: users + 2
Key: 2
Value:
{
  "EndpointAuthorizations": {},
  "Id": 2,
  "Password": "$2a$10$0tfvTx5yBpf0610oqT6G.Nmc8CFB6GP1HVbCBq4MbcfxsQVeV9HS",
  "PortainerAuthorizations": {
    "PortainerDockerHubInspect": true,
    "PortainerEndpointExtensionAdd": true,
    "PortainerEndpointExtensionRemove": true,
    "PortainerEndpointGroupList": true,
    "PortainerEndpointInspect": true,
    "PortainerEndpointIntList": true,
    "PortainerExtensionList": true,
    "PortainerMOTD": true,
    "PortainerRegistryInspect": true,
    "PortainerRegistryList": true,
    "PortainerTeamList": true,
    "PortainerTemplateInspect": true,
    "PortainerTemplateList": true,
    "PortainerUserInspect": true,
    "PortainerUserList": true,
    "PortainerUserMemberships": true
  },
  "Role": 1,
  "Username": "soupeLadmin"
}

```

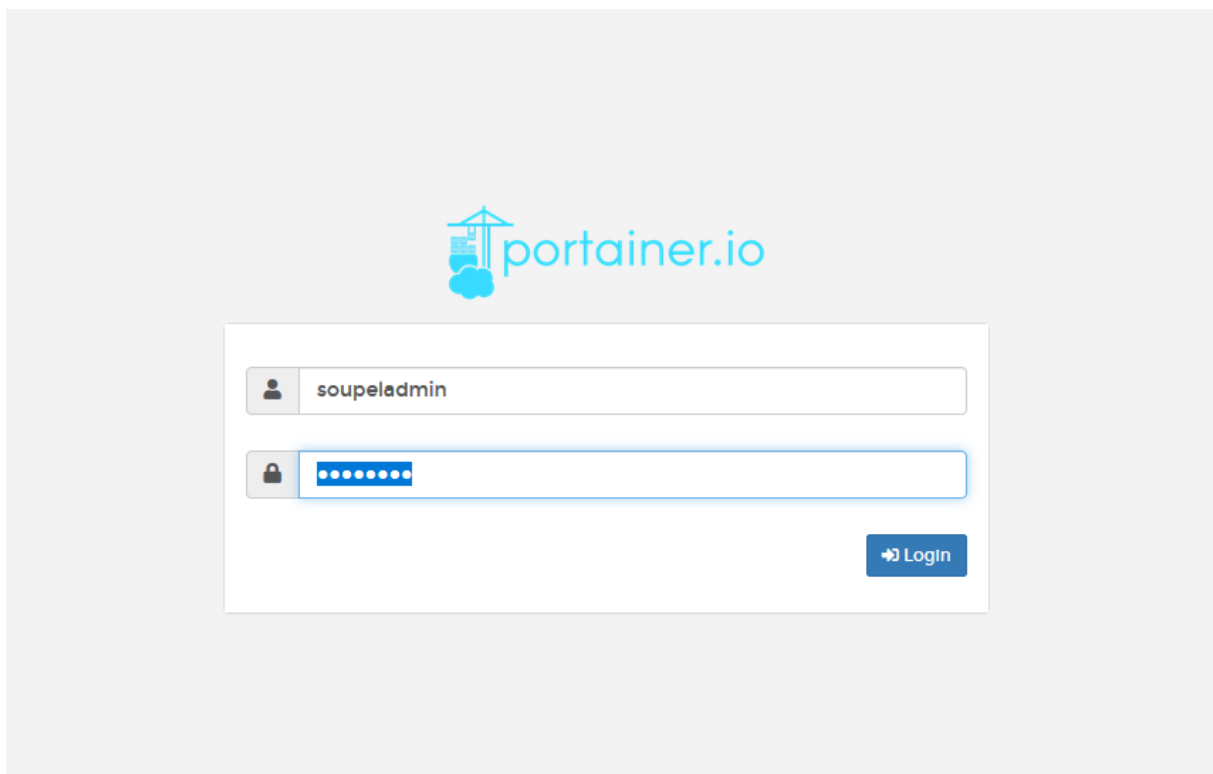
Press enter on the user

```

+-----+
+               Input new value for '      '
+-----+
+
+ |"Id":2,"Username":"soupeLadmin","Password": "  ", "Role":1,"PortainerAuthorizations":{"Portaine
+
+-----+

```

Past your hash here, press Q and restart your docker,



## Portainer recover admin password (BRUTE)

***docker stop portainer***

***docker volume prune***


***docker rm (ID container bloquant)***

***docker volume rm portainer\_data***

***docker volume create portainer\_data***

***docker run -d -p 9000:9000 --name portainer --restart always -v***


***/var/run/docker.sock:/var/run/docker.sock -v portainer\_data:/data portainer/portainer-ce***



Please create the initial administrator user.

**Username**

**Password**

**Confirm password**  

✖ The password must be at least 8 characters long

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



# Install Gitlab with docker

You need to stop the Gitea container before to install Gitlab.

Then you can run the following command :

```
sudo docker run --detach --hostname gitlab.example.com --publish 443:443 --publish 80:80 --publish 44:22 --name gitlab --restart always --volume $GITLAB_HOME/config:/etc/gitlab --volume $GITLAB_HOME/logs:/var/log/gitlab --volume $GITLAB_HOME/data:/var/opt/gitlab gitlab/gitlab-ee:latest
```

The initialization process may take a long time, you can track this process with this command :

```
sudo docker logs -f gitlab
```

## Gitlab runner

To install the gitlab runner you need to run the following command :

```
docker run -d --name gitlab-runner --restart always -v /srv/gitlab-runner/config:/etc/gitlab-runner -v /var/run/docker.sock:/var/run/docker.sock gitlab/gitlab-runner:latest
```

Then to configure it run :

```
docker run --rm -it -v /srv/gitlab-runner/config:/etc/gitlab-runner gitlab/gitlab-runner register
```

And go to your gitlab web site in admin area, runners, and you will find the URL and the Registration Token for the runner.

⚠ Open registration is enabled on your instance.

[Learn more](#) about how you can customize / disable registration on your instance.

[View setting](#)

Admin Area > Runners

Runners are processes that pick up and execute CI/CD jobs for GitLab. You can register runners as separate users, on separate servers, and on your local machine. Register as many runners as you want.

Runners can be:

- **shared** - Runs jobs from all unassigned projects.
- **group** - Runs jobs from all unassigned projects in its group.
- **specific** - Runs jobs from assigned projects.
- **locked** - Cannot be assigned to other projects.
- **paused** - Not available to run jobs.

Set up a shared runner manually

1. Install [GitLab Runner](#) and ensure it's running.
2. Register the runner with this URL:  
`http://192.168.0.48/`

And this registration token:  
`jxsDyqr2yLq4iLauTfdH`

[Reset registration token](#)

[Show Runner installation instructions](#)

Recent searches ▾

Search or filter results...

Created date ▾

Runners currently online: 0

No runners found

```

Enter the GitLab instance URL (for example, https://gitlab.com/):
http://192.168.0.48/
Enter the registration token:
jxsDyqr2yLq4iLauTfdH
Enter a description for the runner:
[29a3d0620c09]: RUNNER_NSA
Enter tags for the runner (comma-separated):
NSA
Registering runner... succeeded runner=jxsDyqr2
Enter an executor: ssh, docker-ssh+machine, kubernetes, docker-ssh, shell, parallels, virtualbox, docker+machine, custom, docker:
docker
Enter the default Docker image (for example, ruby:2.6):
ruby:2.6
Runner registered successfully. Feel free to start it, but if it's running already the config should be automatically reloaded!

```

Refresh your browser and you will get something like that :

Admin Area > Runners

Runners are processes that pick up and execute CI/CD jobs for GitLab. You can register runners as separate users, on separate servers, and on your local machine. Register as many runners as you want.

Runners can be:

- shared** - Runs jobs from all unassigned projects.
- group** - Runs jobs from all unassigned projects in its group.
- specific** - Runs jobs from assigned projects.
- locked** - Cannot be assigned to other projects.
- paused** - Not available to run jobs.

**Set up a shared runner manually**

1. Install [GitLab Runner](#) and ensure it's running.
2. Register the runner with this URL:  
`http://192.168.0.48/`

And this registration token:  
`jxsDyqr2yLq4iLauTfdH`

[Reset registration token](#)

[Show Runner installation instructions](#)

Recent searches  Search or filter results... Created date  Runners currently online: 1

Type/State	Runner token	Description	Version	IP Address	Projects	Jobs	Tags	Last contact	
<b>shared</b> <b>locked</b>	4s2tSjuz	RUNNER_NSA	13.9.0	172.17.0.1	n/a	0	NSA	in 59 minutes	<a href="#">edit</a> <a href="#">stop</a> <a href="#">delete</a>

You need to edit your fresh pipeline by clicking on the pen left to the redcross.

**This runner processes jobs for all unassigned projects.**

If you want a runner to build only specific projects, restrict the project in the table below. After you restrict a runner to a project, you cannot change it back to a shared runner.

Active ☒ Paused runners don't accept new jobs

Protected ☐ This runner will only run on pipelines triggered on protected branches

Run untagged jobs ☒ Indicates whether this runner can pick jobs without tags

Lock to current projects ☐ When a runner is locked, it cannot be assigned to other projects

IP Address

Description

Maximum job timeout   
Enter the number of seconds, or other human-readable input, like "1 hour". This timeout takes precedence over lower timeouts set for the project.

Tags   
You can set up jobs to only use runners with specific tags. Separate tags with commas.

[Save changes](#)

**Restrict projects for this runner**

Project
<input type="text"/> <a href="#">Search</a>

GitLab Instance / Monitoring [Enable](#)

**Recent jobs served by this runner**

Job	Status	Project	Commit	Finished at
-----	--------	---------	--------	-------------

You need to uncheck the “lock to current projects” and check the “Run untagged jobs” if you want it to work.

# How to install NRPE AGENT (Nagios)

## Client Side

### Supported Distributions

The Linux agent installation is currently supported on RHEL/CentOS/Oracle Linux/CloudLinux 5+, Fedora14+, SLES 11+, OpenSUSE 11+, Ubuntu 12+, and Debian 6+.

### Installing The Agent

Download the Linux NRPE agent to the /tmp directory on the Linux server you wish to monitor.

```
cd /tmp
```

```
wget https://assets.nagios.com/downloads/nagiosxi/agents/linux-nrpe-agent.tar.gz
```

Unpack the installation archive you just downloaded:

```
tar xzf linux-nrpe-agent.tar.gz
```

Enter the newly created agent subdirectory:

```
cd linux-nrpe-agent
```

Run the wrapper script **as root** (if using *Ubuntu* you'll need to either run `sudo -i` to run as root or run the

command with `sudo` in front):

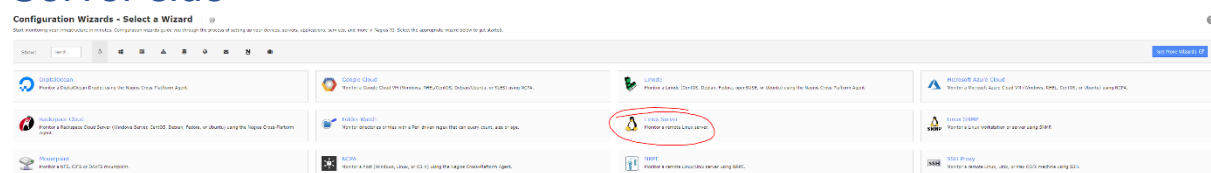
```
./fullinstall
```


This will automatically take care of a number of things for you, including:

- Modifying the distro's package manager repositories
- Installing prerequisite packages
- Creating required users and groups
- Defining services for xinetd
- Compiling and installing the agent and plugins
- Configuring the firewall
- Configuring the agent


At the end of the execution of the script, you need to enter the **IP address** of the **Nagios server**.

## Server side





## Configuration Wizard: Linux Server - Step 1



Linux Server Information

IP Address:

192.168.0.28

The IP address or FQDNS name of the Linux server you'd like to monitor.

Linux Distribution:

RedHat Enterprise

CentOS

Fedora

Oracle

Ubuntu

Debian

SUSE Enterprise

OpenSUSE

Other

ing on the server you'd like to monitor.

< Back

Next >

Enter the IP address of your target VM and enter the distribution.

After that you can add a lot of parameters to monitor your linux VM. You can add warning and critical threshold. To not generate a lot of warnings you are free to add your own values.




## Configuration Wizard: Linux Server - Step 2



### Linux Server Details

IP Address:

Operating System:   
CentOS

Host Name:   
The name you'd like to have associated with this Linux server.

### Linux Agent

You will need to install an agent on the Linux server in order to monitor its metrics.

Agent Download:  [Download Agent](#)

Agent Install Instructions:  [Agent Installation Instructions](#)



SSL Encryption:



Determines whether or not data between the Nagios XI server and Linux agent is encrypted.  
**Note:** Legacy NRPE installations may require that SSL support be disabled.



### Server Metrics



Specify which services you'd like to monitor for the Linux server.



- ☒ **Ping**  
Monitors the server with an ICMP ping. Useful for watching network latency and general uptime.
- ☒ **Yum Update Status**  
Monitors the server to ensure it's up to date with the latest RPM packages.
- ☒ **Load**  
Monitors the load on the server (1,5,15 minute values).  



  
- ☒ **CPU Statistics**  
Monitors the server CPU statistics (% user, system, iowait, and idle)  

  %   %
- ☒ **Memory Usage**  
Monitors the memory usage on the server.  

  %   %
- ☒ **Swap Usage**  
Monitors the swap usage on the server.  

  %   %
- ☒ **Open Files**  
Monitors the number of open files on the server.  

  
- ☒ **Users**  
Monitors the number of users currently logged in to the server.  

  
- ☒ **Total Processes**  
Monitors the total number of processes running on the server.

You can monitor services, but you can do it on the configuration wizards

#### Services

Specify any services normally started by the init process that should be monitored to ensure they're in a running state.

init.d Service	Display Name
<input checked="" type="checkbox"/> sshd	SSH Server
<input checked="" type="checkbox"/> crond	Cron Scheduling Daemon
<input type="checkbox"/> syslog	System Logging Daemon
<input type="checkbox"/> httpd	Apache Web Server
<input type="checkbox"/> mysqld	MySQL Server
<input type="checkbox"/> sendmail	Sendmail Mail Transfer Agent
<input type="checkbox"/> dovecot	Dovecot Mail Server

[Add Row](#) | [Delete Row](#)

#### Processes

Specify any process names that should be monitored to ensure they're running.

Linux Process	Display Name
<input type="checkbox"/> sendmail	Sendmail
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	

[Add Row](#) | [Delete Row](#)

You can deepen check the problems by configuring "When a potential problem is first detected"



## Configuration Wizard: Linux Server - Step 3



### Monitoring Settings

Define basic parameters that determine how the host and service(s) should be monitored.

#### Under normal circumstances:

Monitor the host and service(s) every  minutes.

#### When a potential problem is first detected:

Re-check the host and service(s) every  minutes up to  times before sending a notification.

[< Back](#)

[Next >](#)

[✓ Finish](#)

# Install Grafana & Prometheus

## Grafana install with docker

Run the following command :

```
docker run -d -p 3000:3000 grafana/grafana
```

If you want to run on another port grafana you can launch the following command:

```
docker run -d --name=grafana -p 3456:3000 grafana/grafana
```

After that you need to go on the IP address of your web browser, enter admin and a random password, it will ask you to modify it and it's over.

## Prometheus install with Node Explorer

With Prometheus you can monitor multiple services, here we will explain to you how to monitor a VM with Node Explorer:

First you need to create a file, prometheus.yml at /tmp/ then paste this.

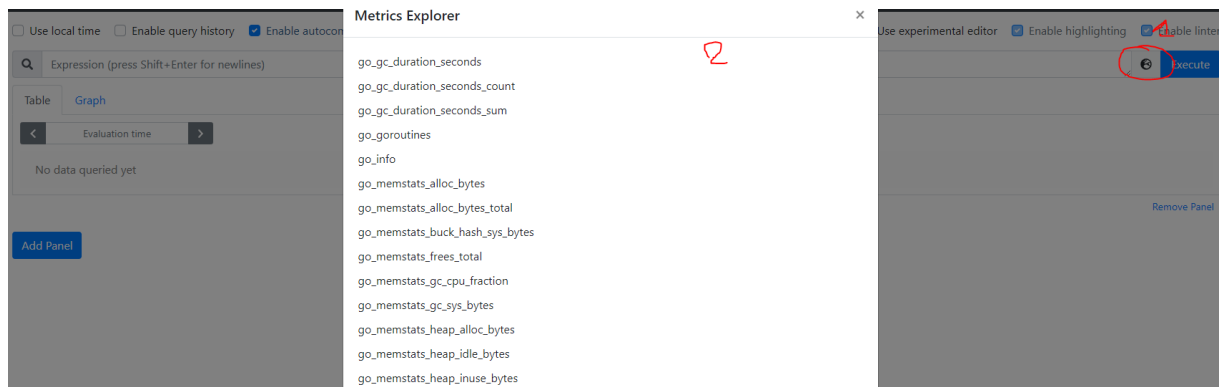
Nano /tmp/prometheus.yml

```
"global:
  scrape_interval: 5s
  external_labels:
    monitor: 'node'
scrape_configs:
  - job_name: 'prometheus'
    static_configs:
      - targets: ['xxx.xxx.xxx.xxx:9090'] ## IP Address of the localhost
  - job_name: 'node-exporter'
    static_configs:
      - targets: ['xxx.xxx.xxx.xxx:9100'] ## IP Address of the localhost"
```

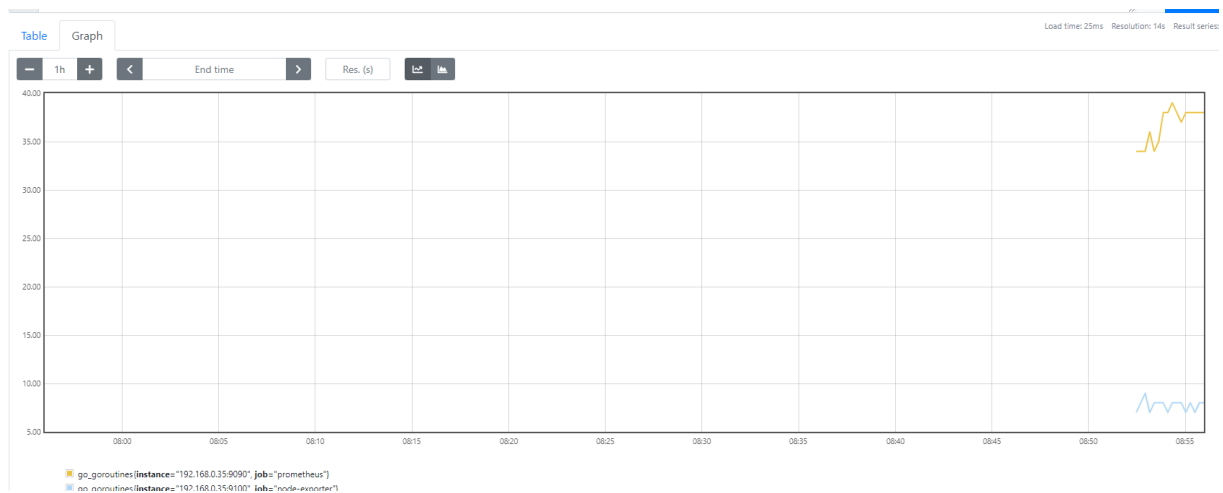
After that you need to run :

```
docker run -d --name prometheus -p 9090:9090 -v
/tmp/prometheus.yml:/etc/prometheus/prometheus.yml prom/prometheus
```

When you are on prometheus you can select the request that you need. And execute it.



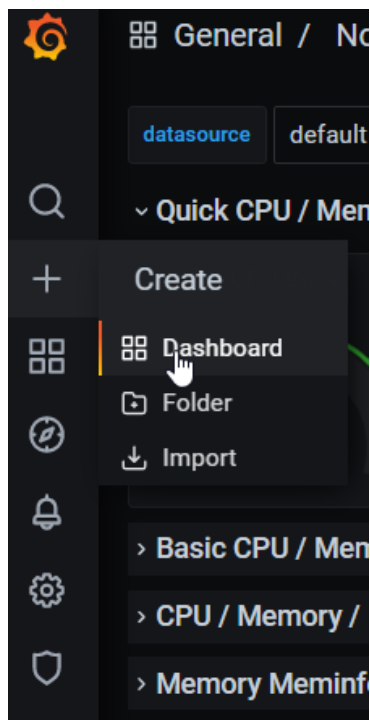
After that you can select “Graph” and you will get a graph of the command how’s running.



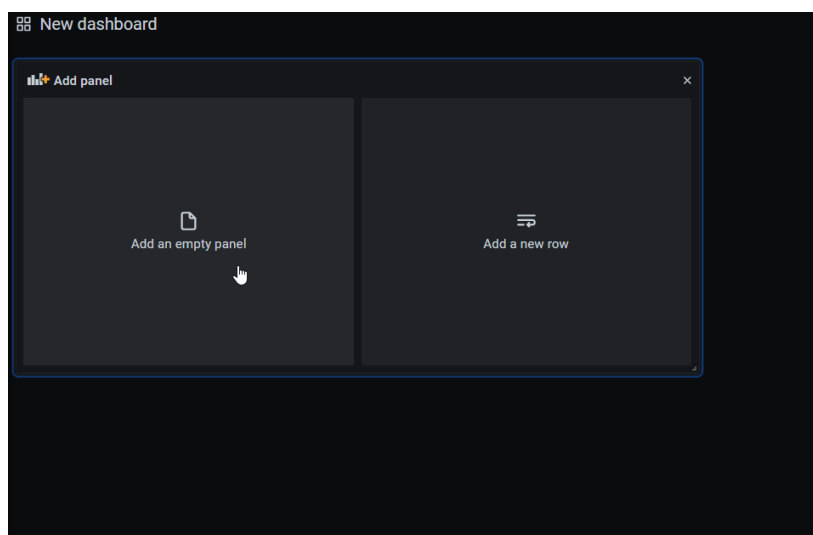


Now we will link prometheus to grafana :

Click on the +, then dashboard :

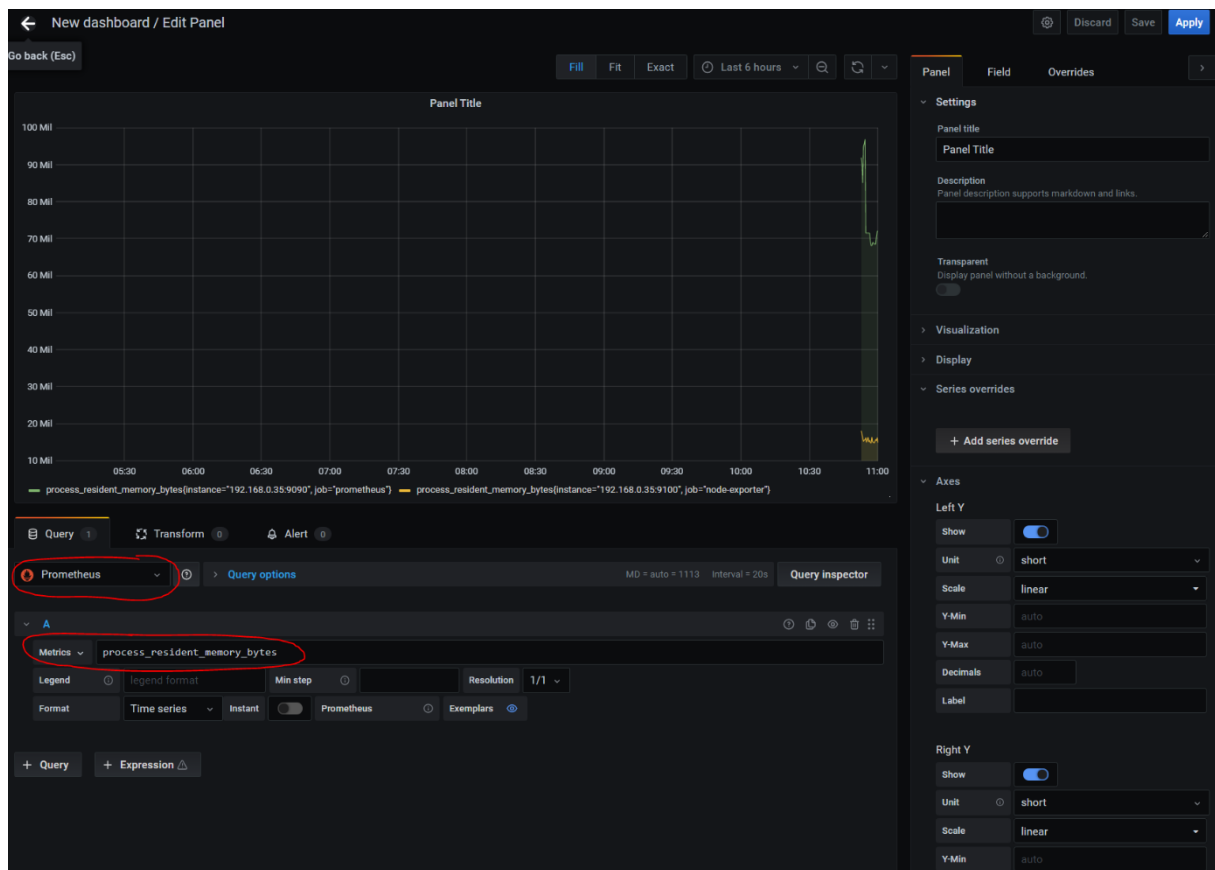


Then add an empty panel :

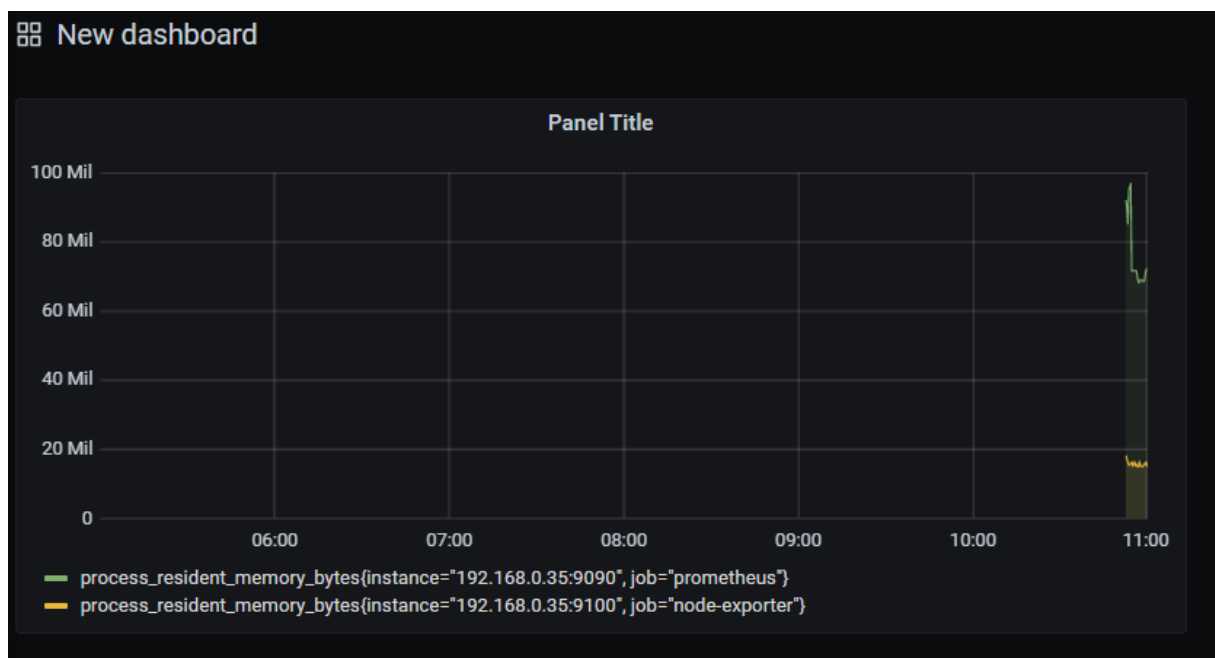


And you will get a new page with a lot of parameters, you need to select prometheus, and the metrics that you need to supervise.

You can add X metrics by clicking on Query at the bottom of the page.



After that you will get this :



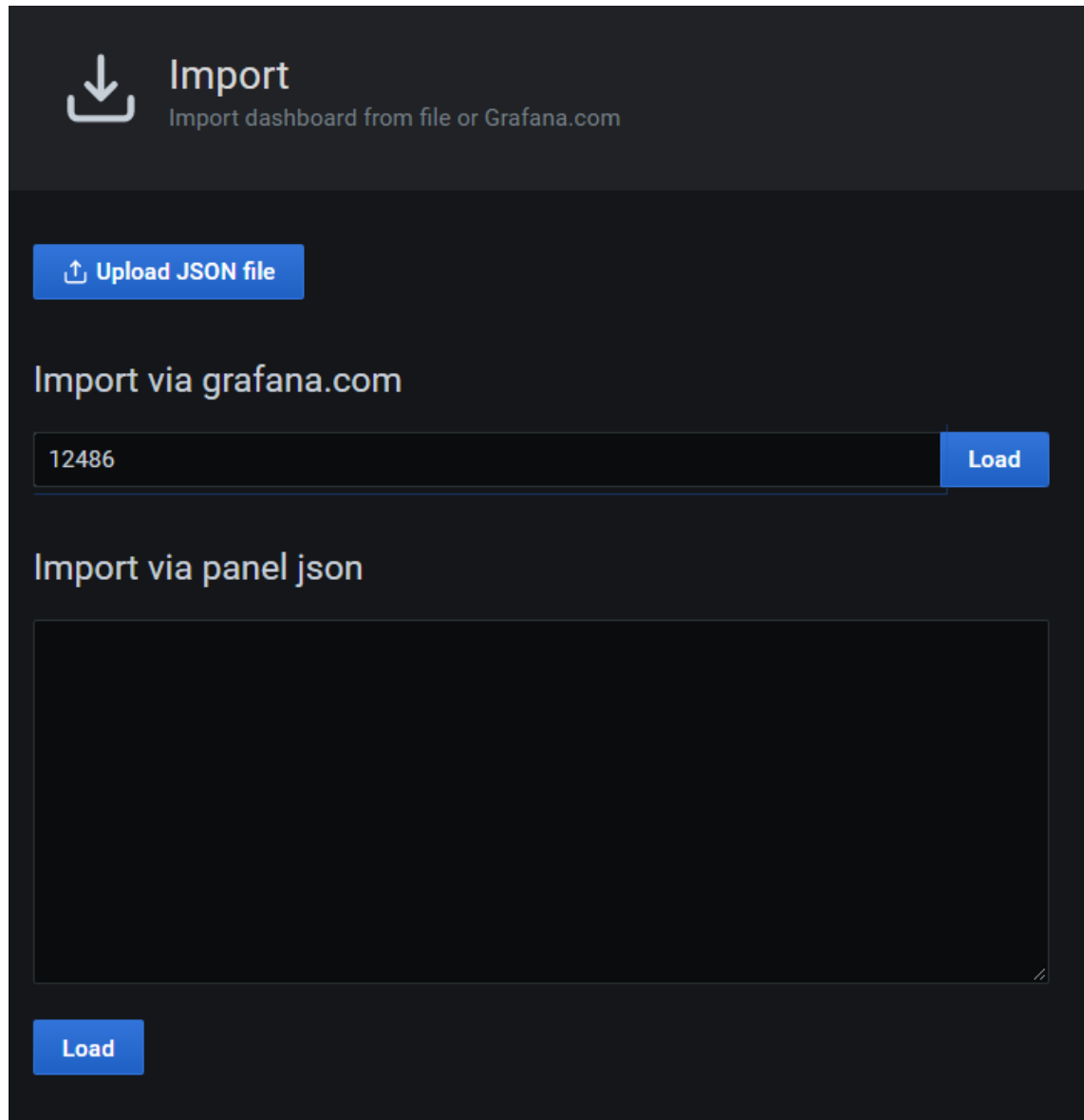
Your panel with the specifics metrics running.

## Import specific Dashboard

You can also import specific Dashboard, by clicking on +, then import. You will have something like that. You need to enter the number of the dashboard, of it's URL :


<https://grafana.com/grafana/dashboards>

Here we will use the Dashboard number "12486", Then click on Load.



The image shows the Grafana 'Import' interface. At the top, there is a header with a download icon and the text 'Import' and 'Import dashboard from file or Grafana.com'. Below this, there is a blue button labeled 'Upload JSON file'. The main section is titled 'Import via grafana.com' and contains a text input field with the value '12486' and a blue 'Load' button. Below this, there is a section titled 'Import via panel json' with a large text area for pasting JSON and a blue 'Load' button at the bottom left.

If your dashboard already exist, you will get the red warnings, but here it's not important, you will then have to click on "import" and you will get your new dashboard



## Import

Import dashboard from file or Grafana.com


### Importing Dashboard from Grafana.com

Published by	candlerb
Updated on	2020-06-22 17:04:51

### Options

Name

Node Exporter Full

 A dashboard or a folder with the same name already exists

Folder


General

Unique identifier (uid)

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

rYdddIPWj

Change uid

 Dashboard named 'Node Exporter Full' in folder 'General' has the same uid

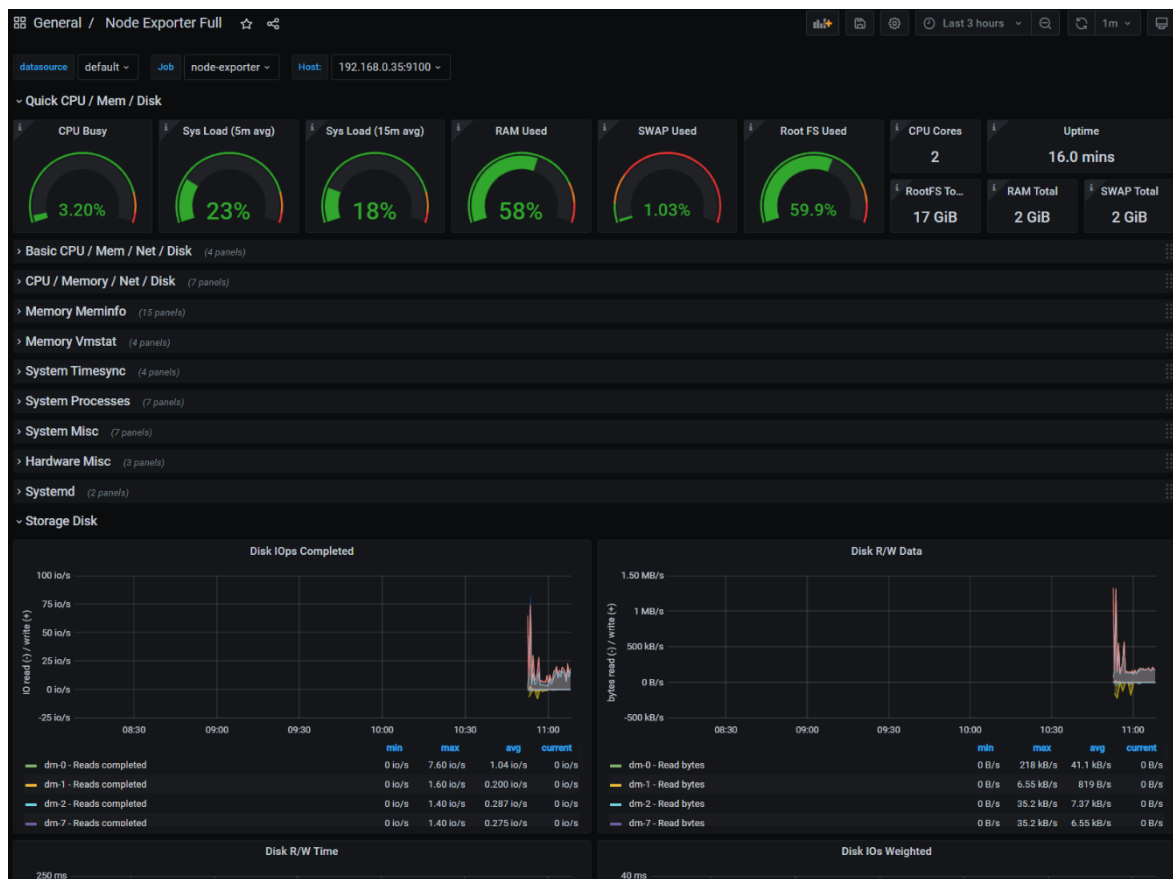
prometheus

Select a Prometheus data source

Import (Overwrite)

Cancel

Normally will get something like that.



## Install Mysql Exporter

First you need to go in your database :

```
docker exec -it <db_container> mysql -uroot -p
```

And create the following user :

```
mysql> CREATE USER 'exporter'@'%' IDENTIFIED BY 'exporterpassword' WITH  
MAX_USER_CONNECTIONS 3;  
mysql> GRANT PROCESS, REPLICATION CLIENT, SELECT ON *.* TO 'exporter'@'%';  
Then run :
```

```
docker run -d \  
--name mysql80-exporter \  
--publish 9104 \  
--restart always \  
--env DATA_SOURCE_NAME="exporter:exporterpassword@(<IP_ADDRESS>:3306)/" \  
prom/mysql80-exporter:latest \  
--collect.info_schema.processlist \  
--collect.info_schema.innodb_metrics \  
--collect.info_schema.tablestats \  
--collect.info_schema.tables \  
--collect.info_schema.userstats \  
--collect.engine_innodb_status
```

After that you need to modify the prometheus.yml file with the following lines :

```
- job_name: 'mysql'
  static_configs:
    - targets: ['192.168.0.35:32770']
```

Then you need to stop the prometheus container and erase it, after that you need to run :

```
docker run \
  -p 9090:9090 \
  -v /tmp/prometheus.yml:/etc/prometheus/prometheus.yml \
  prom/prometheus
```

normally you will get something like this on prometheus.

mysql (1/1 up) [show less](#)

Endpoint	State	Labels
<a href="http://192.168.0.35:32770/metrics">http://192.168.0.35:32770/metrics</a>	UP	instance="192.168.0.35:32770" job="mysql"

If you want to import the dashboard related of mysql-exporter in grafana, the number of the dashboard is 7362.



# Firewall configuration

## Matrice de Flux

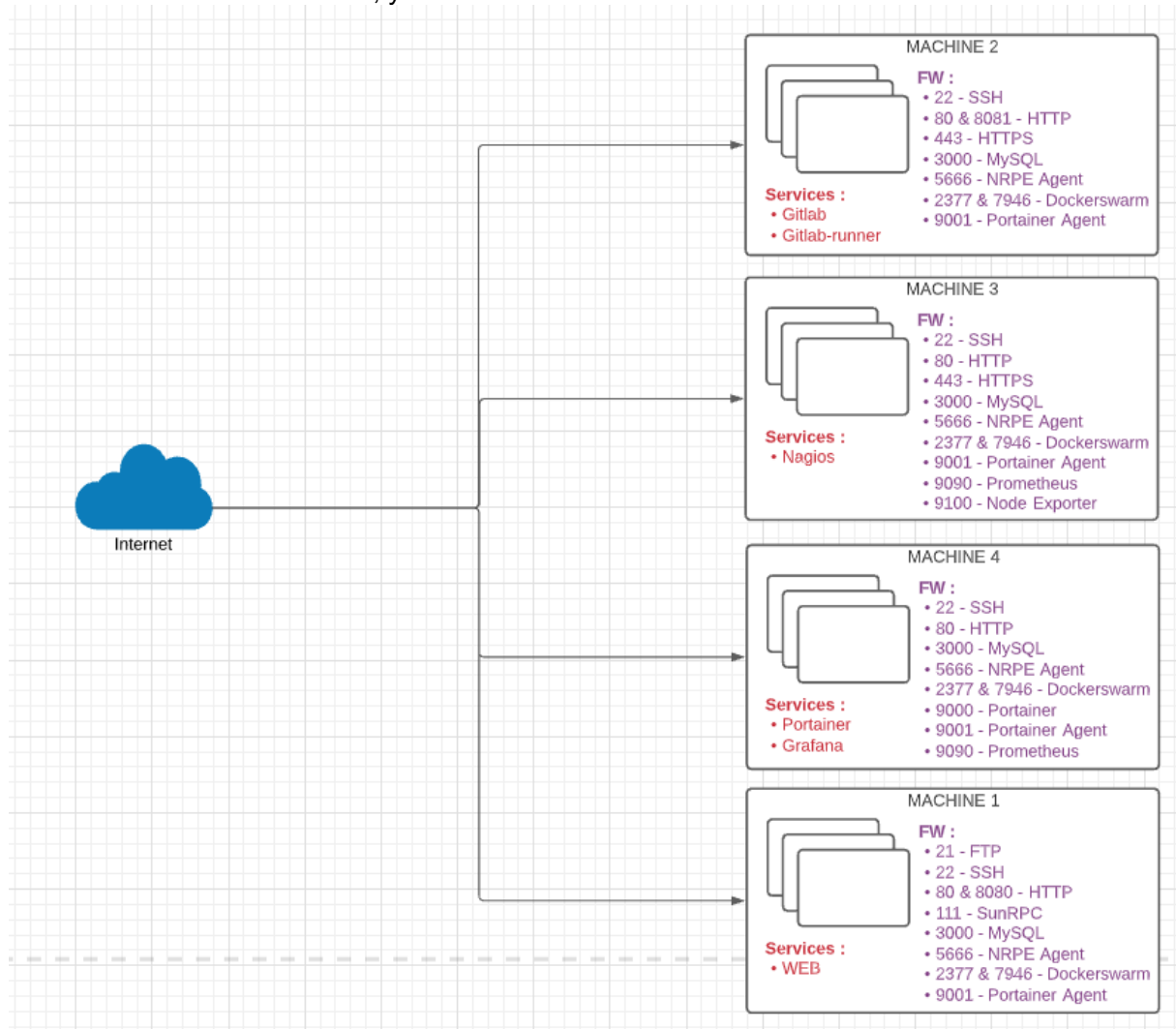
Firstly, you need to install "lsof" packager, to launch the following command :

`lsof -i -P -n | grep LISTEN.`

And you will get the following output :

```
[root@localhost ~]# lsof -i -P -n | grep LISTEN.
sshd          944 root    3u      IPv4    27456      0t0  TCP *:22 (LISTEN)
sshd          944 root    4u      IPv6    27462      0t0  TCP *:22 (LISTEN)
xinetd       953 root    5u      IPv6    27478      0t0  TCP *:5666 (LISTEN)
master      1134 root   13u      IPv4    28288      0t0  TCP 127.0.0.1:25 (LISTEN)
master      1134 root   14u      IPv6    28289      0t0  TCP [::1]:25 (LISTEN)
dockerd     1185 root   29u      IPv6    31550      0t0  TCP *:2377 (LISTEN)
dockerd     1185 root   39u      IPv6    32310      0t0  TCP *:7946 (LISTEN)
docker-pr   1545 root    4u      IPv4    29488      0t0  TCP *:9000 (LISTEN)
docker-pr   2518 root    4u      IPv4    38734      0t0  TCP *:9001 (LISTEN)
docker-pr  17569 root    4u      IPv4   128689      0t0  TCP *:3000 (LISTEN)
docker-pr 115416 root    4u      IPv4   703607      0t0  TCP *:80 (LISTEN)
[root@localhost ~]#
```

With the different information, you can make a schema like this :



## Installation

```
sudo yum install firewalld
sudo systemctl enable firewalld
sudo systemctl start firewalld
sudo firewall-cmd --state
```

## Flow Blockage

### Suppression of services and ports

```
firewall-cmd - -permanent --remove-port=444/tcp
firewall-cmd - -permanent --remove-service=mysql
```

### Add an exception

#### Add a service or a port in exception

```
firewall-cmd --permanent --add-port=22/TCP
firewall-cmd - -permanent --add-service=https
```

## Restart FirewallD

You need to restart the firewall if you want it to take effect.

```
firewall-cmd --reload
```

## Revert to default (for reset)

```
revert to default ( pour reset de base ) firewall-cmd --set-target=default --zone=public fi
firewall-cmd --reload
```



## Docker restart always

`docker update --restart unless-stopped <container_name>`

## NTP synchronized : yes

```
systemctl stop ntpd
ntpd -gq
systemctl start ntpd
```

## Update root password on mysql

If we want to show every user, we need to run the following command :

```
select * from mysql_user;
```

We changed all the passwords of the different root user with this following commands:

```
ALTER USER 'root'@'%' IDENTIFIED BY 'password';
ALTER USER 'root'@'172.18.0.1' IDENTIFIED BY 'password';
ALTER USER 'root'@'localhost' IDENTIFIED BY 'password';
```

We saw 3 different root on 3 different hosts, so we changed everything.

```
mysql> select Host User from mysql.user;
+-----+
| User |
+-----+
| %    |
| 172.18.0.1 |
| localhost |
| localhost |
| localhost |
+-----+
5 rows in set (0.00 sec)

mysql> select Host User Password from mysql.user;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right
syntax to use near 'Password from mysql.user' at line 1
mysql> select User from mysql.user;
+-----+
| User |
+-----+
| root |
| root |
| mysql.session |
| mysql.sys |
| root |
+-----+
5 rows in set (0.00 sec)

mysql> ALTER USER 'root'@'%' IDENTIFIED BY 't5JrMOU0Cbr5zl0syXf0';
Query OK, 0 rows affected (0.00 sec)

mysql> ALTER USER 'root'@'172.18.0.1' IDENTIFIED BY 't5JrMOU0Cbr5zl0syXf0';
Query OK, 0 rows affected (0.00 sec)

mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY 't5JrMOU0Cbr5zl0syXf0';
Query OK, 0 rows affected (0.00 sec)

mysql>
```

# Change a user password

You can run the following command to see every users:

cat /etc/passwd

```
[root@localhost ~]# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
avahi-autoipd:x:170:170:Avahi IPv4LL Stack:/var/lib/avahi-autoipd:/sbin/nologin
systemd-bus-proxy:x:999:997:systemd Bus Proxy:/:/sbin/nologin
systemd-network:x:998:996:systemd Network Management:/:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
polkitd:x:997:995:User for polkitd:/:/sbin/nologin
tss:x:59:59:Account used by the trousers package to sandbox the tcsd daemon:/dev/null:/sbin/nologin
postfix:x:89:89:/:/var/spool/postfix:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/ssh:/sbin/nologin
admin:x:1000:1000:admin:/home/admin:/bin/bash
dockerroot:x:996:993:Docker User:/var/lib/docker:/sbin/nologin
service-web:x:1002:1002:/:/home/service-web:/bin/bash
nagios:x:1003:1003:/:/home/nagios:/bin/bash
saslauth:x:995:76:Saslauthd user:/run/saslauthd:/sbin/nologin
gitlab-runner:x:1004:1005:GitLab Runner:/home/gitlab-runner:/bin/bash
```

You will see every users by the following characteristic “**/bin/bash**”

If you want to change the password of a user, you need to run the following command :

***sudo -u <username> passwd // OR // sudo passwd <username> Docker restart always***