

Installation - STEP 1 - System preparation

DO NOT USE ROOT FOR INSTALLATION. SUDO ONLY

Main server installation

You can use virtual & physical machine. WSL is also supported

Server installation

[Install Ubuntu 20.04](#) (almost the same for newer versions)

Update system

```
sudo apt update
sudo apt dist-upgrade
sudo reboot
```

[Set up your firewall](#)

Set your current timezone on server

```
sudo tzselect
```

Install needed software support

```
sudo apt install snmp nmap traceroute fping snmp-mibs-downloader git wget unzip curl
```

To prevent error (Cant get all info or Error "Unknown Object Identifier (Index out of range: XXX (ifindex))" on Huawei OLT and other devices we recommended to disable RangeCheck:

```
sudo nano /etc/snmp/snmp.conf
```

Set config to this:

```
# As the snmp packages come without MIB files due to license reasons, loading
# of MIBs is disabled by default. If you added the MIBs you can reenale
# loading them by commenting out the following line.
#mibdirs /usr/share/snmp/mibs:/usr/share/snmp/mibs/iana:/usr/share/snmp/mibs/ietf
#mibs ALL
mibs :

# If you want to globally change where snmp libraries, commands and daemons
# look for MIBs, change the line below. Note you can set this for individual
# tools with the -M option or MIBDIRS environment variable.
#
# mibdirs /usr/share/snmp/mibs:/usr/share/snmp/mibs/iana:/usr/share/snmp/mibs/ietf
noRangeCheck yes
```

Install PHP and modules

```
sudo apt install lib-release ca-certificates apt-transport-https software-properties-common -y

sudo add-apt-repository ppa:ondrej/php

sudo apt install php8.2-[cli,common,curl,fpm,gd,gmp,imagick,intl,mbstring,memcache,mysql,opcache,snmp,xsl,xslrpc]

sudo update-alternatives --set php /usr/bin/php8.2
```

Installing Nginx as webserver

```
sudo add-apt-repository universe
sudo apt update
sudo apt install nginx
sudo ufw allow 'Nginx HTTP'
sudo ufw allow 'Nginx HTTPS'
```

Edit php.ini for Ngnix

```
sudo nano /etc/php/8.2/fpm/php.ini
```

and change this line

```
file_uploads = On
allow_url_fopen = On
memory_limit = 1024M
upload_max_filesize = 8M
max_execution_time = 600
max_input_time = 300 ; or more for low speed devices
max_input_vars = 20000 ; It is important
post_max_size = 8M
default_charset = "UTF-8"
date.timezone = Europe/Kyiv
```

Edit php.ini for CLI

```
sudo nano /etc/php/8.2/cli/php.ini
```

and change this line

```
memory_limit = 1024M
max_execution_time = 1800
max_input_time = 300 ; or more for low speed devices
max_input_vars = 20000 ; It is important
default_charset = "UTF-8"
date.timezone = Europe/Kyiv
```

See also <https://wintercms.com/docs/v1.2/docs/setup/configuration>

Restart nginx and php8.2-fpm

```
sudo systemctl restart nginx
sudo systemctl restart php8.2-fpm.service
```

Fine PHP-FPM tuning

You can see configuration example [here](#) or search on GOOGLE like "How to optimize php-fpm" 🍷

Installing Memcached

```
sudo apt install memcached libmemcached-tools php8.2-memcached php-memcache
```

Edit memcached.conf

```
sudo nano /etc/memcached.conf
```

Change **-m 64** to **-m 128 -I 16M**

```
# Start with a cap of 64 megs of memory. It's reasonable, and the daemon default
# Note that the daemon will grow to this size, but does not start out holding this much
# memory
-m 128 -I 16M
```

-m 128 - This is memcache storage size
-I 16m - This is item cache size (default 1 Megabyte)

```
sudo systemctl restart memcached
sudo systemctl enable memcached
```

If you have large network (more then 10 000 devices or 200 000 ONUs) you can increase value to **-m 256 -I 32M**

Installing Maria DB

Installing Maria DB server

You can install default version or latest

Installing version from Ubuntu repository (recommended for small network)

```
sudo apt install mariadb-server mariadb-client
```

OR install latest for big or large network (see version [here](#)). At this moment latest was [11.8.2](#)

Skip next if you are installing (or have installed) the default version of Mariadb

```
sudo apt install software-properties-common -y
curl -LsS -O https://downloads.mariadb.com/MariaDB/mariadb_repo_setup
```

For Ubuntu 20.04

```
sudo bash mariadb_repo_setup --os-type=ubuntu --os-version=focal --mariadb-server-version=11.7
```

For Ubuntu 22.04

```
sudo bash mariadb_repo_setup --os-type=ubuntu --os-version=jammy --mariadb-server-version=11.7
```

For Ubuntu 24.04

```
sudo bash mariadb_repo_setup --os-type=ubuntu --os-version=noble --mariadb-server-version=11.7
```

Update repository and install Maria DB

```
sudo apt update
sudo apt install mariadb-server mariadb-client
```

Start and enable server

```
sudo systemctl start mariadb
sudo systemctl enable mariadb
```

Secure MYSQL

Run this command and prevent instructions

```
sudo mariadb-secure-installation
```

Set up user and DB

This is default login and password to database. You can set any what you want and then set it in config

```
mariadb -u root -p
MariaDB [(none)]> CREATE DATABASE grusher;
MariaDB [(none)]> GRANT ALL ON grusher.* TO 'grusher'@'localhost' IDENTIFIED BY 'grusher' WITH GRANT OPTION;
MariaDB [(none)]> FLUSH PRIVILEGES;
MariaDB [(none)]> quit;
```

Installing Python virtual enviroment

Ubuntu 18.04 - 22.04 (better to use Python 3.11)

You can install a version no lower than 3.11

```
sudo apt install python3.11 python3-venv python3-pip python3-virtualenv python3.11-distutils
```

If you have error with previous command - try next commands and when repeat previous command

```
sudo add-apt-repository ppa:deadsnakes/ppa -y
sudo apt update
```

Installing other component

```
/usr/bin/python3 -m pip install --upgrade pip
sudo mkdir /opt/python3.11
sudo mkdir /opt/python3.11/env
sudo chown -R $USER:www-data /opt/python3.11/
virtualenv --python=/usr/bin/python3.11 /opt/python3.11/env/
cd /opt/python3.11/env/bin/
source /opt/python3.11/env/bin/activate
./python3 -m pip install --upgrade pip
./python3 -m pip install --upgrade wheel
./python3 -m pip install --upgrade git+https://github.com/gviabcua/netmiko.git
./python3 -m pip install --upgrade ping3
./python3 -m pip install --upgrade requests
./python3 -m pip install --upgrade psutil
./python3 -m pip install --upgrade zipp
deactivate
```

Then in Grusher settings check python path

Default is set to `/opt/python3.12/env/bin/python3`

See here [GRUSHER_IP/system/config?option_type=search_param&search_data=python](#)

Ubuntu 24.04 + (better to use Python 3.12)

You can install a version no lower than 3.12

```
sudo apt install python3.12 python3-venv python3-pip python3-virtualenv
```

Installing other component

```
/usr/bin/python3 -m pip install --upgrade pip
sudo mkdir /opt/python3.12
sudo mkdir /opt/python3.12/env
sudo chown -R $USER:www-data /opt/python3.12/
virtualenv --python=/usr/bin/python3.12 /opt/python3.12/env/
cd /opt/python3.12/env/bin/
source /opt/python3.12/env/bin/activate
./python3 -m pip install --upgrade pip
./python3 -m pip install --upgrade wheel
./python3 -m pip install --upgrade git+https://github.com/gviabcua/netmiko.git
./python3 -m pip install --upgrade ping3
./python3 -m pip install --upgrade requests
./python3 -m pip install --upgrade psutil
./python3 -m pip install --upgrade zipp
deactivate
```

If you have error like this `ImportError: cannot import name 'html5lib' from 'pip_vendor'` run this:

```
curl -sS https://bootstrap.pypa.io/get-pip.py | python3.11
```

Then in Grusher settings check python path

Default is set to `/opt/python3.11/env/bin/python3`

See here [GRUSHER_IP/system/config?option_type=search_param&search_data=python](#)

Installing Composer2

```
cd
php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
php composer-setup.php
php -r "unlink('composer-setup.php');"
sudo mv composer.phar /usr/local/bin/composer
```