1 Installing Nightscout on Ubuntu 20.04 LTS server (on DigitalOcean or MVPS)

find a German version of this guide here:[1]

1.1 Why?

- You have full SSH access to your server
- Creating a server takes a minute
- You can choose from various places on earth where the server should be located
- You can make a backup / snapshot of the server in order to delete and later restore it (saves money if you don't need it for a while)
- You pay per hour, the basis version at $5 (DigitalOcean) or or 3 EUR (MVPS) per month are sufficient for Nightscout
- Loads of documentation on how to install things
- Control and track the server via the DigitalOcean or MVPS control panel website or alternatively remote via ssh

1.2 Step-by-Step

You can download a pdf version this guide here[2].

1.2.1 Create Droplet on DigitalOcean or a VPS on MVPS

DigitalOcean is probably the better known provider and they offer a wide range of services. But I personally prefer MVPS.

DigitalOcean

- Sign up or login at DigitalOcean[3]
- Click Create a Droplet
- Under Distributions, select Ubuntu 20.04 LTS
- Under Size, the smallest 5 $ version is sufficient (but we will have to temporarily resize it during the installation process. Don't try to start with the 10$, downsizing to 5$ won't work)
- The rest is up to you. Click Create to finish the process.

[1] <https://gist.github.com/DrCR77/11b5698fc304b9a214dfa15862f3722b>
[2] <http://docs.c-rathmann.de/NSonDOorMVPS.pdf>
[3] <https://m.do.co/c/896e26961756>

MVPS

- In a similar way you can create a VPS on MVPS[4]
- The basis account for only 3 EUR per month (less than 30 EUR p.a. with annual payment method) is sufficient (all prices +VAT).
- MVPS uses European server locations

1.2.2 Login to your Droplet/VPS

Open a Terminal or whatever you use to get SSH access to a Server. If you are new to Linux or haven't heard about SSH before, don't give up! It is easy and you will find an good reference here[5].
Use the provided Login data you got sent via Mail to connect to the Droplet/VPS (I'll use *123.456.789.123* as a replacement for the actual IP of your Droplet):

```
ssh root@123.456.789.123
// enter password when asked for it
```

1.2.3 Install and Setup MongoDB

MongoDB has to be installed and configured to be (a) secured and (b) provide the database for Nightscout.

```
sudo apt-get update && apt-get upgrade -y
sudo apt install mongodb -y
```

Secure MongoDB

```
mongo --port 27017
```

You should have entered the mongo shell. We now create an admin user, remember what you picked for *ADMIN_NAME* and *ADMIN_PASSWORD*. Type the following (just press Enter before inserting the next line, copy-paste should work as well):

```
use admin
db.createUser({user: "ADMIN_NAME", pwd: "ADMIN_PASSWORD", roles: [ { role: "userAdminAnyDatabase", db: "admin" }] })
```

[4] <https://www.mvps.net/?aff=17519>
[5] <https://www.howtogeek.com/311287/how-to-connect-to-an-ssh-server-from-windows-macos-or-linux/>

The shell should return this code and Successfully added user.
Exit the Mongo shell (STRG+C) and type:

```
sudo nano /etc/mongodb.conf
```

Find the line where it says:

```
#auth = true
```

and remove the #
(Save and exit a file that you opened with nano by STRG+O, Enter, STRG+X)
Restart Mongo:

```
sudo service mongodb restart
```

Create database for Nightscout

Now we log back into the Mongo shell as admin and create our Nightscout database where *MONGO*NS*DB* is the name for the database we create, *MONGO*NS*USER* is the user who can access this database and *MONGO*NS*PASSWORD* is his password. Adjust user and password to your needs (src[6]):

```
mongo -u ADMIN_NAME -p --authenticationDatabase admin
// enter ADMIN_PASSWORD
```

In the Mongo shell enter the following:

```
use MONGO_NS_DB
db.createUser({user: "MONGO_NS_USER", pwd: "MONGO_NS_PASSWORD", roles: [ { role: "readWrite", db: "MONGO_NS_DB" }]})
```

The shell should return Successfully added user. Exit the Mongo shell again via CTRL+C or by typing exit.
You should be able to login as the user you just created:

```
mongo -u MONGO_NS_USER -p --authenticationDatabase MONGO_NS_DB
// enter MONGO_NS_PASSWORD
```

exit (STRG+C)

[6] <https://docs.mongodb.com/manual/tutorial/create-users/>

1.2.4 Create a non-root user

Add a non-root user with admin rights. Don't try to run nightscout as root user, it will now work. Replace mainuser with a name of your choice.

```
sudo adduser mainuser
```

After you've set and verified the password, you don't need to put any more information, just keep on hitting enter. Next, we'll be adding the new user to the administrative users list:

```
sudo usermod -aG sudo mainuser
```

Now, we're going to login to the machine using the account we just created.

```
ssh mainuser@IP-number
```

1.2.5 Install Nightscout

Install nodejs and npm:

```
curl -sL https://deb.nodesource.com/setup_12.x | sudo -E bash -
sudo apt-get install nodejs -y
```

Clone Nightscout and create a folder link with the name "nightscout" for easier reference:

```
git clone https://github.com/nightscout/cgm-remote-monitor.git
sudo ln -s cgm-remote-monitor nightscout
cd nightscout
```

Now install Nightscout:

```
npm install
```

If you chose the $5/month plan for the Droplet, this will possibly fail with the message killed. The reason is missing RAM. That's why we have to temporarily resize the Droplet to a 10$-plan just to install Nightscout.

- via SSH, type shutdown now to turn off the Droplet

- On the DigitalOcean website, go to Droplets
- Open your Nightscout Droplet
- Go to Resize (menu on the left side)
- Click on the $10 plan and wait until the process is done
- On the top right corner, there is a slider saying Off, click on the slider to turn the Droplet On
- Login again via SSH and repeat these steps:

  ```
  cd cgm-remote-monitor
  npm install
  ```

  Afterwards you can turn the Droplet off again, go back to the DigitalOcean site and revert the plan (select the $5 plan, turn the Droplet on, reconnect via SSH).

1.2.6 Configure

Edit the config file:

```
nano my.env
```

Insert the following and further keys as needed:

```
MONGO_CONNECTION=mongodb://MONGO_NS_USER:MONGO_NS_PASSWORD@127.0.0.1:27017/MONGO_NS_DB
BASE_URL=http://123.456.789.123:1337
DISPLAY_UNITS=mg
DEVICESTATUS_ADVANCED="true"
mongo_collection="entries"
API_SECRET=NIGHTSCOUT_API_SECRET
ENABLE=careportal%20openaps%20iob%20bwp%20sage%20cage%20basal%20pump%20maker
```

*NIGHTSCOUT*API*SECRET* is something long you can pick yourself.
The BASE_URL consists of the Droplet IP and the Port 1337 on which the Nightscout app runs (= make sure to attach :1337 to your IP there).
In my case I had to connect different values with %20 instead of spaces and " to make it work.
Save and close again (STRG+ O, Enter, STRG+ X).

1.3 Startup - Install pm2 to monitor nightscout process

```
sudo npm install pm2 -g
```

Start cgm-remote-monitor with pm2:

```
env $(cat my.env)  PORT=1337 pm2 start server.js
```

Make pm2 start cgm-remote-monitor on startup

```
pm2 startup
```

This will give you a command you need to run as superuser to allow pm2 to start the app on reboot.
The command will be something like: sudo su -c "env PATH=$PATH:/usr/bin pm2 startup ..."
Copy the tailored command, paste and execute it. Finally in order to make pm2 remember what apps to start on next reboot:
pm2 save
Nightscout can now be started, stopped and restarted like this (you don't have to do this now):

```
pm2 start server
pm2 stop server
pm2 restart server
```

Please note that every time you make modifications to your my.env file (your Nightscout environment variables), you will have to stop, delete and restart the server.js instance. Go to your /cgm-remote-monitor folder and:
pm2 list will show your current processes, and their id number.
pm2 stop 0 will stop the current server process with id "0".
pm2 delete 0 will delete the server process with id "0".
You also need to use pm2 unstartup systemd and repeat the startup process again.

```
pm2 cleardump
pm2 save
```

Then start cgm-remote-monitor with pm2 again: env $(cat my.env) PORT=1337 pm2 start server.js
I noticed that the server didn't start up correctly after a reboot/power off/on. The reason could be the boot sequence calling for pm2 before the MongoDB is up and running. A simple entry in the crontab solves this issue:

```
crontab -e
#add this line to the mainuser crontab:
@reboot pm2 restart server | at now + 2 minutes
```

Alternatively you could use a shell script (see "troublechecksu.sh" below) after startup to check if Nightscout indicates "trouble" on the website. It then restarts the server via pm2 and this should solve the issue.
I added this as root use to the crontab as calling it from crontab -e (mainuser crontab) didn't work:

```
sudo nano /etc/crontab
###add this line
@reboot root  /home/mainuser/backup/troublechecksu.sh
```

1.3.1 Test Nightscout

You might be able to use Nightscout already. This is the URL (same as BASE_URL in my.env):

```
http://123.456.789.123:1337
```

Most browsers don't allow this unsecure connection and hence you should follow the next steps.

1.3.2 Setup secure connection (SSL / HTTPS)

Since it is very recommended to not use HTTP, but HTTPS for your Nightscout instance, I add the steps to get a HTTPS address. You need a domain for this - domains are only a few dollars a year - or you use a subdomain of a domain you already own. This is what I did. To attach the domain to your Droplet, follow the DigitalOcean guides on How to setup Domain[7], How to setup a subdomain[8], or How to add Domains from common registrars to Droplet[9]. The process is only slightly different in MVPS, an A-record pointing to the IP-address and 2 NS entries are needed on your domain/subdomain.
To secure the site and make it work in secured state, install nginx:

```
sudo apt install certbot python3-certbot-nginx -y
```

Configure nginx to use your domain:

[7] <https://www.digitalocean.com/community/tutorials/how-to-set-up-a-host-name-with-digitalocean>
[8] <https://www.digitalocean.com/community/tutorials/how-to-set-up-and-test-dns-subdomains-with-digitalocean-s-dns-panel>
[9] <https://www.digitalocean.com/community/tutorials/how-to-point-to-digitalocean-nameservers-from-common-domain-registrars>

```
sudo nano /etc/nginx/sites-available/default
```

Then I had to do the followoing 2 steps.
1) Initially I added MYDOMAIN to the default configuration:

```
server {
    listen 80;
    ...
    server_name MYDOMAIN.COM;
    ...
```

Test if the configuration file is valid via sudo nginx -t. Afterwards, restart nginx:

```
sudo service nginx restart
```

You have to enable ufw first by

```
sudo ufw allow ssh && sudo ufw allow 22
sudo ufw enable
```

(You can confirm with y as ssh on port 22 has been allowed).
Configure the firewall ufw to open the neccessary ports (use sudo ufw app list to check that nginx is visible to ufw as an app):

```
sudo ufw allow 'Nginx Full' && sudo ufw allow 443 && sudo ufw allow 1337 && sudo ufw allow 1337/tcp
sudo ufw status
sudo service nginx restart
```

Obtaining an SSL Certificate with this command (answer: a(gree) ; no ; 2):

```
sudo certbot --nginx -d MYDOMAIN.COM
```

improve SSL security by generating a strong Diffie-Hellman group (src[10]):

```
sudo openssl dhparam -out /etc/ssl/certs/dhparam.pem 2048
```

[10] <https://gist.github.com/johnmales/1b3c927f2a56aae640b4b2cd0298b1e7>

Make sure certbot takes care of renewing your certificate:

```
sudo certbot renew --dry-run
```

Restart the service:

```
sudo service nginx restart
```

You can check how secure your site is via https://www.ssllabs.com/ssltest/index.html.
2) At this point I got several website errors (redirection etc) and didn't get a website response. So I had to reconfigure nginx default file again:

```
sudo nano /etc/nginx/sites-available/default
```

I had to add these lines in the location section of the 443 server:

```
server {

    listen [::]:443 ssl ipv6only=on; # managed by Certbot
    listen 443 ssl; # managed by Certbot

     server_name MYDOMAIN.COM;

... (a few lines managed by Certbot)

location / {

      proxy_set_header        Host $host;
      proxy_set_header        X-Real-IP $remote_addr;
      proxy_set_header        X-Forwarded-For $proxy_add_x_forwarded_for;
      proxy_set_header        X-Forwarded-Proto $scheme;

      # Fix the \It appears that your reverse proxy set up is broken" error.
      proxy_pass          http://localhost:1337;
      proxy_read_timeout  90;
```

```
      proxy_redirect      http://localhost:1337 https://MYDOMAIN.COM;
    }
}
```

Nightscout should now be accessible via

```
https://MYDOMAIN.COM
```

If you run into errors like "502 Bad Gateway" the process likely failed because pm2 installation failed and NS server is not up. To test the Nightscout installation try it without pm2:

```
cd nightscout
screen

sudo env $(cat my.env) PORT=1337 node server.js
```

You can leave the screen with CTRL+A CTRL+D and reopen it with screen -x or kill the process with STRG+C

1.4 Backup and Restore Mongo-Database via mongodump

In the basic subscription of a droplet you can take a snapshot of your working droplet whenever you like. To store one snapshot would cost about 20 cents per month. Alternatively you could add automatic weekly backups for 1 $ per month.
You might want to backup and/or transfer the mongo-database to another droplet. In order to do this follow the next steps:

1.4.1 Create a backup

Login as mainuser to the server (via ssh or console). Create a backup folder:

```
sudo mkdir backup
```

Now you are ready to execute this command:

```
sudo mongodump --user MONGO_NS_USER --password MONGO_NS_PASSWORD --db MONGO_NS_DB --out backup/
```

But it is better to stop the server first, create a backup and restart the server. Thus, use the script backup.shbelow to run this process.

1.4.2 Restoring a backup

```
sudo mongorestore --db=MONGO_NS_DB --username=MONGO_NS_USER  --password=MONGO_NS_PASSWORD backup/MONGO_NS_DB
```

To avoid data inconsistencies in the MongoDB it is advisable to stop the nightscout process before executing mongodump/mongorestore. You can use below stated shell scripts within the backup directory that stops the process via pm2, executes backup or restore and then restarts the process again.
Create a file by sudo nano backup.sh
Copy and paste below mentioned content into the script and exit via CTRL+x and confirm saving.
Do the same with sudo nano restore.sh
If you want to create backups with a daily/weekly routine I suggest to add a line to the root users crontab:

```
sudo  nano /etc/crontab
```

and add a line pointing to an adjusted backup script (see below backupsu.sh):

```
0 9    * * *        root    /home/mainuser/backup/backupsu.sh
```

Make all scripts executable by sudo chmod +x *.sh
Please note that both scripts assume that security authentification of the MONGO*NS*DB was enabled as described above.
(This guide was originally forked from https://gist.github.com/frauzufall/c69f4a76730e3eb24e7a582d636765df and adjusted.)

1.4.3 Extras and Trouble Shooting

To install a slim version of ubuntu you can start the installatioin with this command [https://ostechnix.com/debfoster-keep-only-essential-packages-in-debian-and-ubuntu/]:

```
sudo apt-get --simulate purge $(dpkg-query -Wf '${Package;-40}${Priority}\n' | awk '$2 ~ /optional|extra/ { print $1 }')
```

DO or MVPS servers are constantly under attack of bots that try to brute force their login. That means you should always use a password with high entropy but it will also lead to a very big journal log file. Over time this log file might take up several GB or even the entire space of your VPS.
You can view the journal with this command:

```
sudo journalctl -r
```

To reduce an inflated journal you can use these commands:

```
journalctl --vacuum-size=100M
```

This will retain the most recent 100M of data.
But the best solution is to reduce the max usage of the journal [https://askubuntu.com/questions/1238214/big-var-log-journal]:

```
sudo nano /etc/systemd/journald.conf
```

SystemMaxUse=100M
Furthemore, it might be good to install an extra security level like fail2ban:
https://linuxize.com/post/install-configure-fail2ban-on-ubuntu-20-04/
https://upcloud.com/community/tutorials/scan-ubuntu-server-malware/
And ultimately switch to ssh key login and disable the ssh password login: https://ulrike-haessler.de/ssh-schluessel-unter-mac-os-x-2/
If intend to reduce monthly costs of your server in case you don't need the droplet on DO temporarily you can follow this guide:
https://vpsfix.com/6359/digitalocean-reduce-cost-unused-droplets/#:~:text=End%20up%20paying%20for%20full,for%20the%20turned%20off%20droplets.

1.5 Files

shell_backup.sh

```
#!/bin/bash

datum=`date +%Y%m%d`

cd ~
pm2 stop server
sudo mongodump --db MONGO_NS_DB -u MONGO_NS_USER -p MONGO_NS_PASSWORD  --out backup/
#sudo mongodump --db MONGO_NS_DB --out backup/  ## in case no authentification required

cd ~/backup
sudo tar -zcf MONGO_NS_DB_$datum.tar.gz MONGO_NS_DB
sudo rm -r MONGO_NS_DB

cd ~
pm2 restart server
```

shell_backupsu.sh

```
#!/bin/bash
#point to this script from /etc/crontab
datum=`date +%Y%m%d`

cd /home/mainuser
su mainuser -c "pm2 stop server"

sudo mongodump --db MONGO_NS_DB -u MONGO_NS_USER -p MONGO_NS_PASSWORD  --out backup/
#sudo mongodump --db MONGO_NS_DB --out backup/  ## in case no authentification required

cd /home/mainuser/backup
sudo tar -zcf MONGO_NS_DB_$datum.tar.gz MONGO_NS_DB
sudo rm -r MONGO_NS_DB

su mainuser -c "pm2 restart server"
```

shell_restore.sh

```
#!/bin/bash
# Restoring the backup created by mongodump

datum=`date +%Y%m%d`

cd ~/backup
sudo tar -vxzf MONGO_NS_DB_$datum.tar.gz  # This scripts assumes there is a backup available that was
    created today. If not, change this line to your own needs/tar-file name.

cd ~
pm2 stop server
sudo mongorestore --db=MONGO_NS_DB --username=MONGO_NS_USER --password=MONGO_NS_PASSWORD backup/
    MONGO_NS_DB

cd ~/backup
sudo rm -r MONGO_NS_DB

cd ~
pm2 restart server
```

shell_troublechecksu.sh

```
#!/bin/bash

exec 3>&1 4>&2
trap 'exec 2>&4 1>&3' 0 1 2 3
exec 1>/home/mainuser/backup/check.log 2>&1

echo " "
date
sleep 60
date

if curl https://mydomain.com | grep trouble; then
    echo "trouble found"
#    sudo service mongodb start
#    sleep 5
  su mainuser -c "pm2 restart server"
else
    echo "no trouble found"
fi
```