

## Ubuntu v17.10.1 Installation for Mining Rigs:

v 1.10 06/04/2018 © swisstrader

1. Download latest Ubuntu installation file from <https://www.ubuntu.com/download/desktop> and create a bootable USB flash stick with the Ubuntu ISO file. Use the help tool program 'Rufus' to create the bootable USB flash stick.
2. Install Ubuntu v17.10.1 **CHECK**  Download updates while installing Ubuntu  
and **DO NOT CHECK**  Install third-party software for graphics and WiFi hardware, MP3 and other media  
Your name **<user>**: *your\_user\_name*  
Your device name: *your\_device\_name*  
Password: *your\_password*
3. **DO NOT CHECK**  Log in automatically and click on CONTINUE button to start installation process
4. Reboot after installation of Ubuntu v17.10.1. with click on 'Restart Computer' to reboot it, remove the USB Stick after you hear the beep of BIOS.
5. Open "Settings" and deactivate all Power Management settings under 'Power'.
6. Open "Software & Updates" and deactivate all auto update settings in the tab "Updates".  
*Install updates from:* let these three hooks checked  
and below set the following preference to  
*Never*  
*Display immediately*  
*Display immediately*  
*Never*
7. Start "Software Updates" and reboot the machine.
8. For Ubuntu version 17.10.1 to install the NVIDIA-driver you have to install at first:  

```
sudo apt-add-repository multiverse && sudo apt update && sudo apt install nvidia-modprobe
```

The **nvidia-modprobe** utility is used to load NVIDIA kernel modules and create NVIDIA character device files automatically everytime your machine boots up.

Then remove # before lines **WaylandEnable=false** , **AutomaticLoginEnable=true** and **AutomaticLogin=<user>** with editing by  

```
sudo nano /etc/gdm3/custom.conf
```

After then **reboot** the machine.

9. Install the INTEL microcode driver for GPU video Intel HD Graphics 610 (Kaby Lake)  
[ download link: <http://ftp.ubuntu.com/ubuntu/ubuntu/pool/restricted/i/intel-microcode/> ]

```
sudo apt-get install microcode_ctl
sudo apt-get install intel-microcode
OR execute
sudo apt-get install microcode_ctl intel-microcode
```

and **reboot** the machine.

Now check whether the microcode has been loaded properly, with the following terminal command:

```
dmesg | grep microcode
```

10. In Ubuntu systems, drivers for NVIDIA Graphics Cards are already provided in the official repository. Installation is as simple as one command.  
*The NVIDIA link of all existing Linux drivers include CUDA is [http://developer.download.nvidia.com/compute/cuda/repos/ubuntu1704/x86\\_64/](http://developer.download.nvidia.com/compute/cuda/repos/ubuntu1704/x86_64/)*

For ubuntu 16.04.1 LTS, the latest version is 361.

To install the driver execute

```
sudo apt-get install nvidia-361 nvidia-modprobe
```

and then **reboot** the machine.

For ubuntu 17.10.1 LTS, the latest version is 390.

Check the available packages with **sudo apt-get install nvidia-** & press **TAB-key**, the list will be loaded.

To install the driver, execute

```
sudo apt-get install nvidia-390
sudo apt-get install nvidia-modprobe
OR (both together in one command line)
sudo apt-get install nvidia-390 nvidia-modprobe
```

and then **reboot** the machine.

11. To avoid that Ubuntu at booting always overwrites the configuration file **xorg.conf** with its tuning settings back to non-tuned xorg.conf add the attribute '**nogpumanager**' to the file '**grub**' in its settings line GRUB\_CMDLINE\_LINUX\_DEFAULT = "quiet splash **nogpumanager**".

[ see at: <https://askubuntu.com/questions/731990/gpu-manager-overwrites-xorg-conf> ]

For this change open a terminal and execute

```
sudo nano /etc/default/grub
```

Change in the editor the line from

```
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"
```

to

```
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash noppumanager".
```

Close the editor with CTRL+X, (Y)es and ENTER and execute as next

```
sudo update-grub
```

to update the grub configuration file in `/boot/grub/grub.cfg`.

Now reboot the machine to activate changings. Now the file `xorg.conf` will not be overwritten anymore by Ubuntu after a new boot process.

#### PLEASE NOTE !

Go to the directory of `xorg.conf` and make a backup copy of the file inside of the same directory. This backup file might be needed later for editing and reorganizing `xorg.conf` on your system in case of a crash after tuning process made by program '`nvidia-xconfig`'.

```
sudo cp -f /etc/X11/xorg.conf /etc/X11/xorg.conf.backup
```

## 12. Activate the option for TUNING of NVIDIA GPUs

Source: <https://noobminer.xyz/overclocking-multiple-nvidia-graphics-card-linux/#nvidia-smi>

Run one time the following command:

```
sudo nvidia-xconfig --enable-all-gpus --cool-bits=28 --allow-empty-initial-configuration
```

and now save a copy of the recent by program `nvidia-xconfig` changed file `xorg.conf` to the desktop with executing following command line:

```
sudo cp -f /etc/X11/xorg.conf ~/Desktop/xorg.conf.tuning
```

This saved file named to '`xorg.conf.tuning`' might be needed for some adaptations by editing with editor `nano` or a simple text editor.

## Option "Coolbits" "integer"

Source: [http://download.nvidia.com/XFree86/Linux-x86\\_64/384.111/README/xconfigoptions.html](http://download.nvidia.com/XFree86/Linux-x86_64/384.111/README/xconfigoptions.html)

The *Coolbits* value is the sum of its component bits in the binary numeral system. The component bits are:

- **1 (bit 0)** - Enables overclocking of older (pre-Fermi) cores on the *Clock Frequencies* page in *nvidia-settings*.
- **2 (bit 1)** - When this bit is set, the driver will "attempt to initialize SLI when using GPUs with different amounts of video memory".
- **4 (bit 2)** - **Enables manual configuration of GPU fan speed on the *Thermal Monitor* page in *nvidia-settings*.**
- **8 (bit 3)** - **Enables overclocking on the *PowerMizer* page in *nvidia-settings*. Available since version 337.12 for the Fermi architecture and newer.**[\[2\]](#)
- **16 (bit 4)** - Enables overvoltage using *nvidia-settings* CLI options. Available since version 346.16 for the Fermi architecture and newer.[\[3\]](#)

To enable multiple features, add the *Coolbits* values together. For example, to enable overclocking and overvoltage of Fermi cores, set Option "Coolbits" "24"

Check the new created *xorg.conf* with executing

```
sudo nano /etc/X11/xorg.conf
```

The [8 GPU version of \*xorg.conf\*](#) for the special mining mainboard **COLORFUL B250A-BTC PLUS V20** should look like:

```
##
## original Ubuntu path and file name of this file are:
## /etc/X11/xorg.conf
##
## Compare the PCI Bus ID's with your mainboard and adapt them in this xorg.conf file!
## If nvidia-smi table shows PCI ID 0A, then write in the xorg.conf file for the PCI BusID 10 .
##

Section "ServerLayout"
    Identifier "layout"
    Screen 0 "Screen0"
    ## take 'LeftOf', 'RightOf', 'Above' or 'Below' for screen positions
    Screen 1 "Screen1" Below "Screen0"
    Screen 2 "Screen2" LeftOf "Screen1"
    Screen 3 "Screen3" LeftOf "Screen2"
    Screen 4 "Screen4" LeftOf "Screen3"
    Screen 5 "Screen5" LeftOf "Screen4"
    Screen 6 "Screen6" LeftOf "Screen5"
    Screen 7 "Screen7" LeftOf "Screen6"
    Inactive "intel"
EndSection

##GPU INTEL with display
##=====
```

```
Section "Device"
    Identifier "intel"
    Driver     "modesetting"
    BusID      "PCI:0:2:0"
    VendorName "INTEL Corporation"
    BoardName  "INTEL HD Graphics 610"
    Option     "AcceMethod" "None"
EndSection
Section "Screen"
    Identifier "intel"
    Device     "intel"
EndSection

##GPU0
##====
Section "Device"
    Identifier "Device0"
    Driver     "nvidia"
    VendorName "NVIDIA Corporation"
    BoardName  "GeForce GTX 1070 Ti"
    BusID      "PCI:1:0:0" # PCI Bus ID is always decimal number, not a hex number!
EndSection
Section "Screen"
    Identifier "Screen0"
    Device     "Device0"
    Option     "AllowEmptyInitialConfiguration" "True"
    Option     "Coolbits" "28"
    SubSection "Display"
        Depth    24
    EndSubSection
EndSection

##GPU1
##====
Section "Device"
    Identifier "Device1"
    Driver     "nvidia"
    VendorName "NVIDIA Corporation"
    BoardName  "GeForce GTX 1070 Ti"
    BusID      "PCI:2:0:0" # PCI Bus ID is always decimal number, not a hex number!
EndSection
Section "Screen"
    Identifier "Screen1"
    Device     "Device1"
    Option     "AllowEmptyInitialConfiguration" "True"
    Option     "Coolbits" "28"
    SubSection "Display"
        Depth    24
    EndSubSection
EndSection
```

```
EndSubSection
EndSection

##GPU2
##====
Section "Device"
    Identifier "Device2"
    Driver "nvidia"
    VendorName "NVIDIA Corporation"
    BoardName "GeForce GTX 1070 Ti"
    BusID "PCI:3:0:0" # PCI Bus ID is always decimal number, not a hex number!
EndSection
Section "Screen"
    Identifier "Screen2"
    Device "Device2"
    Option "AllowEmptyInitialConfiguration" "True"
    Option "Coolbits" "28"
    SubSection "Display"
        Depth 24
    EndSubSection
EndSection

##GPU3
##====
Section "Device"
    Identifier "Device3"
    Driver "nvidia"
    VendorName "NVIDIA Corporation"
    BoardName "GeForce GTX 1070 Ti"
    BusID "PCI:4:0:0" # PCI Bus ID is always decimal number, not a hex number!
EndSection
Section "Screen"
    Identifier "Screen3"
    Device "Device3"
    Option "AllowEmptyInitialConfiguration" "True"
    Option "Coolbits" "28"
    SubSection "Display"
        Depth 24
    EndSubSection
EndSection

##GPU4
##====
Section "Device"
    Identifier "Device4"
    Driver "nvidia"
    VendorName "NVIDIA Corporation"
    BoardName "GeForce GTX 1070 Ti"
```

```
    BusID      "PCI:5:0:0"          # PCI Bus ID is always decimal number, not a hex number!
EndSection
Section "Screen"
    Identifier  "Screen4"
    Device      "Device4"
    Option      "AllowEmptyInitialConfiguration" "True"
    Option      "Coolbits" "28"
    SubSection  "Display"
        Depth    24
    EndSubSection
EndSection

##GPU5
##====
Section "Device"
    Identifier  "Device5"
    Driver      "nvidia"
    VendorName  "NVIDIA Corporation"
    BoardName   "GeForce GTX 1070 Ti"
    BusID      "PCI:7:0:0"          # PCI Bus ID is always decimal number, not a hex number!
EndSection
Section "Screen"
    Identifier  "Screen5"
    Device      "Device5"
    Option      "AllowEmptyInitialConfiguration" "True"
    Option      "Coolbits" "28"
    SubSection  "Display"
        Depth    24
    EndSubSection
EndSection

##GPU6
##====
Section "Device"
    Identifier  "Device6"
    Driver      "nvidia"
    VendorName  "NVIDIA Corporation"
    BoardName   "GeForce GTX 1070 Ti"
    BusID      "PCI:8:0:0"          # PCI Bus ID is always decimal number, not a hex number!
EndSection
Section "Screen"
    Identifier  "Screen6"
    Device      "Device6"
    Option      "AllowEmptyInitialConfiguration" "True"
    Option      "Coolbits" "28"
    SubSection  "Display"
        Depth    24
    EndSubSection
```

```

EndSection

##GPU7
##====
Section "Device"
    Identifier "Device7"
    Driver "nvidia"
    VendorName "NVIDIA Corporation"
    BoardName "GeForce GTX 1070 Ti"
    BusID "PCI:9:0:0" # PCI Bus ID is always decimal number, not a hex number!
EndSection
Section "Screen"
    Identifier "Screen7"
    Device "Device7"
    Option "AllowEmptyInitialConfiguration" "True"
    Option "Coolbits" "28"
    SubSection "Display"
        Depth 24
    EndSubSection
EndSection

```

The [12 GPU version of xorg.conf](#) for the mainboard **ASRock H110 PRO BTC+** should look like:

```

##
## original Ubuntu path and file name of this file are:
## /etc/X11/xorg.conf
##
## Compare the PCI Bus ID's with your mainboard and adapt them in this xorg.conf file!
## If nvidia-smi table shows PCI ID 0A, then write in the xorg.conf file for the PCI BusID 10 .
##

Section "ServerLayout"
    Identifier "layout"
    Screen 0 "Screen0"
    ## take 'LeftOf', 'RightOf', 'Above' or 'Below' for screen positions
    Screen 1 "Screen1" Below "Screen0"
    Screen 2 "Screen2" LeftOf "Screen1"
    Screen 3 "Screen3" LeftOf "Screen2"
    Screen 4 "Screen4" LeftOf "Screen3"
    Screen 5 "Screen5" LeftOf "Screen4"
    Screen 6 "Screen6" LeftOf "Screen5"
    Screen 7 "Screen7" LeftOf "Screen6"
    Screen 8 "Screen8" LeftOf "Screen7"
    Screen 9 "Screen9" LeftOf "Screen8"
    Screen 10 "Screen10" LeftOf "Screen9"
    Screen 11 "Screen11" LeftOf "Screen10"
    Inactive "intel"
EndSection

```

```
##GPU INTEL with display
##=====
Section "Device"
    Identifier "intel"
    Driver     "modesetting"
    BusID      "PCI:0:2:0"
    VendorName "INTEL Corporation"
    BoardName  "INTEL HD Graphics 610"
    Option     "AccelMethod" "None"
EndSection
Section "Screen"
    Identifier "intel"
    Device     "intel"
EndSection

##GPU0
##====
Section "Device"
    Identifier "Device0"
    Driver     "nvidia"
    VendorName "NVIDIA Corporation"
    BoardName  "GeForce GTX 1070 Ti"
    BusID      "PCI:1:0:0"          # PCI Bus ID is always decimal number, not a hex number!
EndSection
Section "Screen"
    Identifier "Screen0"
    Device     "Device0"
    Option     "AllowEmptyInitialConfiguration" "True"
    Option     "Coolbits" "28"
    SubSection "Display"
        Depth 24
    EndSubSection
EndSection

##GPU1
##====
Section "Device"
    Identifier "Device1"
    Driver     "nvidia"
    VendorName "NVIDIA Corporation"
    BoardName  "GeForce GTX 1070 Ti"
    BusID      "PCI:2:0:0"          # PCI Bus ID is always decimal number, not a hex number!
EndSection
Section "Screen"
    Identifier "Screen1"
    Device     "Device1"
    Option     "AllowEmptyInitialConfiguration" "True"
```

```
Option "Coolbits" "28"
SubSection "Display"
    Depth 24
EndSubSection
EndSection

##GPU2
##====
Section "Device"
    Identifier "Device2"
    Driver "nvidia"
    VendorName "NVIDIA Corporation"
    BoardName "GeForce GTX 1070 Ti"
    BusID "PCI:3:0:0" # PCI Bus ID is always decimal number, not a hex number!
EndSection
Section "Screen"
    Identifier "Screen2"
    Device "Device2"
    Option "AllowEmptyInitialConfiguration" "True"
    Option "Coolbits" "28"
    SubSection "Display"
        Depth 24
    EndSubSection
EndSection

##GPU3
##====
Section "Device"
    Identifier "Device3"
    Driver "nvidia"
    VendorName "NVIDIA Corporation"
    BoardName "GeForce GTX 1070 Ti"
    BusID "PCI:4:0:0" # PCI Bus ID is always decimal number, not a hex number!
EndSection
Section "Screen"
    Identifier "Screen3"
    Device "Device3"
    Option "AllowEmptyInitialConfiguration" "True"
    Option "Coolbits" "28"
    SubSection "Display"
        Depth 24
    EndSubSection
EndSection

##GPU4
##====
Section "Device"
    Identifier "Device4"
```

```
Driver      "nvidia"
VendorName  "NVIDIA Corporation"
BoardName   "GeForce GTX 1070 Ti"
BusID       "PCI:5:0:0"          # PCI Bus ID is always decimal number, not a hex number!
EndSection
Section "Screen"
Identifier  "Screen4"
Device     "Device4"
Option     "AllowEmptyInitialConfiguration" "True"
Option     "Coolbits" "28"
SubSection "Display"
    Depth   24
EndSubSection
EndSection

##GPU5
##====
Section "Device"
Identifier  "Device5"
Driver     "nvidia"
VendorName "NVIDIA Corporation"
BoardName  "GeForce GTX 1070 Ti"
BusID      "PCI:6:0:0"          # PCI Bus ID is always decimal number, not a hex number!
EndSection
Section "Screen"
Identifier  "Screen5"
Device     "Device5"
Option     "AllowEmptyInitialConfiguration" "True"
Option     "Coolbits" "28"
SubSection "Display"
    Depth   24
EndSubSection
EndSection

##GPU6
##====
Section "Device"
Identifier  "Device6"
Driver     "nvidia"
VendorName "NVIDIA Corporation"
BoardName  "GeForce GTX 1070 Ti"
BusID      "PCI:9:0:0"          # PCI Bus ID is always decimal number, not a hex number!
EndSection
Section "Screen"
Identifier  "Screen6"
Device     "Device6"
Option     "AllowEmptyInitialConfiguration" "True"
Option     "Coolbits" "28"
```

```
        SubSection    "Display"
            Depth      24
        EndSubSection
    EndSection

##GPU7
##====
Section "Device"
    Identifier    "Device7"
    Driver        "nvidia"
    VendorName    "NVIDIA Corporation"
    BoardName     "GeForce GTX 1070 Ti"
    BusID         "PCI:10:0:0"          # PCI Bus ID is always decimal number, not a hex number!
EndSection
Section "Screen"
    Identifier    "Screen7"
    Device        "Device7"
    Option        "AllowEmptyInitialConfiguration" "True"
    Option        "Coolbits" "28"
    SubSection    "Display"
        Depth      24
    EndSubSection
EndSection

##GPU8
##====
Section "Device"
    Identifier    "Device8"
    Driver        "nvidia"
    VendorName    "NVIDIA Corporation"
    BoardName     "GeForce GTX 1070 Ti"
    BusID         "PCI:11:0:0"          # PCI Bus ID is always decimal number, not a hex number!
EndSection
Section "Screen"
    Identifier    "Screen8"
    Device        "Device8"
    Option        "AllowEmptyInitialConfiguration" "True"
    Option        "Coolbits" "28"
    SubSection    "Display"
        Depth      24
    EndSubSection
EndSection

##GPU9
##====
Section "Device"
    Identifier    "Device9"
    Driver        "nvidia"
```

```
VendorName "NVIDIA Corporation"
BoardName "GeForce GTX 1070 Ti"
BusID "PCI:12:0:0" # PCI Bus ID is always decimal number, not a hex number!
EndSection
Section "Screen"
Identifier "Screen9"
Device "Device9"
Option "AllowEmptyInitialConfiguration" "True"
Option "Coolbits" "28"
SubSection "Display"
Depth 24
EndSubSection
EndSection

##GPU10
##=====
Section "Device"
Identifier "Device10"
Driver "nvidia"
VendorName "NVIDIA Corporation"
BoardName "GeForce GTX 1070 Ti"
BusID "PCI:13:0:0" # PCI Bus ID is always decimal number, not a hex number!
EndSection
Section "Screen"
Identifier "Screen10"
Device "Device10"
Option "AllowEmptyInitialConfiguration" "True"
Option "Coolbits" "28"
SubSection "Display"
Depth 24
EndSubSection
EndSection

##GPU11
##=====
Section "Device"
Identifier "Device11"
Driver "nvidia"
VendorName "NVIDIA Corporation"
BoardName "GeForce GTX 1070 Ti"
BusID "PCI:14:0:0" # PCI Bus ID is always decimal number, not a hex number!
EndSection
Section "Screen"
Identifier "Screen11"
Device "Device11"
Option "AllowEmptyInitialConfiguration" "True"
Option "Coolbits" "28"
SubSection "Display"
```

Depth 24  
EndSubSection  
EndSection

To make the xorg.conf changes stick on reboot execute the lines  
(**444**: root, user and groups have only 'read' permissions AND **+i**: changes the attribute of file to 'immutable') as follows:

```
sudo chmod 444 /etc/X11/xorg.conf && sudo chattr +i /etc/X11/xorg.conf
```

Some people are commenting after a system reboot, the changes do not stick. One way to avoid this is to set the permission of the conf file to **read-only ( 444 ) and immutable ( +i )** (make sure you jot down the permissions before in case you encounter problems).

**Please note for later changings in the file xorg.conf:**

To remove the 'freeze' of file **xorg.conf** execute:

```
sudo chattr -i /etc/X11/xorg.conf && sudo chmod 644 /etc/X11/xorg.conf
```

The tuning option of NVIDIA GPUs is permanently active now, **reboot** the machine!

13. Open a Terminal and run \$ **sudo apt install gdebi-core** to install helptool package Gdebi (see Website at <https://launchpad.net/gdebi> ) and reboot the PC by running \$ **reboot** in the terminal. This tool helps to install \*.deb files.

14. Open a Terminal and run

```
$ sudo apt-get update           # Fetches the list of available updates
$ sudo apt-get upgrade         # Strictly upgrades the current packages
$ sudo apt-get dist-upgrade    # Installs updates (new ones)
$ sudo reboot                  # Reboots machine
```

to update Ubuntu to the most recent software versions include security updates.

15. Open “Settings” and change the display resolution in ‘Devices => Displays’ to 1920 x 1080 (16:9).

16. Open [www.teamviewer.com](http://www.teamviewer.com) and download latest installation file i386 x64 Debian/Ubuntu of Teamviewer for Ubuntu/Debian. Open windows ‘Files’ from favorites task bar and then go to the subfolder ‘Downloads’. Double-click on the teamviewer\_xx.x.xxxx\_i386.deb file and then click on ‘Install’ button. **Reboot the machine**. After then start Teamviewer, go to Options => icon Security to set a your Personal Password and Random password (for spontaneous access) to ‘Password strength => Disabled (no random password). Teamviewer is now prepared for later use to control the miner.

17. Click on icon ‘Show Applications’ (*Nine Points square-icon left side down on desktop*) and add ‘Teamviewer’ and other icons to favorites task bar.

|                             |  |  |
|-----------------------------|--|--|
| 18. Open a terminal and run | \$ <b>sudo apt-get install nano</b>            | <b>//simple text editor</b>  |
|                             | \$ <b>sudo apt-get install nvidia-settings</b> | <b>//NVIDIA package to adapt settings</b>                                    |
|                             | \$ <b>sudo apt-get install nvidia-smi</b>      | <b>//start it with <i>watch -n 0.2 ('1' stands for 1 sec) nvidia-smi</i></b> |
|                             | \$ <b>sudo apt-get install ssh</b>             | <b>//remote controll</b>   |
|                             | \$ <b>sudo apt-get install stress</b>          | <b>//imposes certain types of compute stress on system</b>                   |
|                             | \$ <b>sudo apt-get install tree</b>            | <b>//lists contents of directories in a tree-like format</b>                 |
|                             | \$ <b>sudo apt-get install vim</b>             | <b>//improved text editor</b>  |

To add more helpful programs of ‘top’ group open a terminal and run following command lines:

|  |  |
|--|--|
| \$ <b>sudo apt-get install htop</b>      | <b>//tests the CPU</b>   |
| \$ <b>sudo apt-get install iftop</b>     | <b>//checks all data transfer and connections via network</b>          |
| \$ <b>sudo apt-get install iotop</b>     | <b>//checks input and output of disks</b>                              |
| \$ <b>sudo apt-get install net-tools</b> | <b>//installs net tools like ‘ifconfig’ and ‘ip addr’</b>              |
| \$ <b>sudo apt-get install dcfldd</b>    | <b>//disk dump (dd) with progress indication (block progress info)</b> |
| \$ <b>sudo apt-get install parted</b>    | <b>//resize (enlarge or decrease) partition</b>                        |

19. **Repair** by editing with editor 'nano' the following **Ubuntu v17.10.1 bug**:

```
sudo nano /etc/gdm3/custom.conf
```

The screenshot shows a terminal window with the nano editor open to the file `/etc/gdm3/custom.conf`. The file content is as follows:

```
GNU nano 2.8.6 File: /etc/gdm3/custom.conf
GDM configuration storage
#
# See /usr/share/gdm/gdm.schemas for a list of available options.

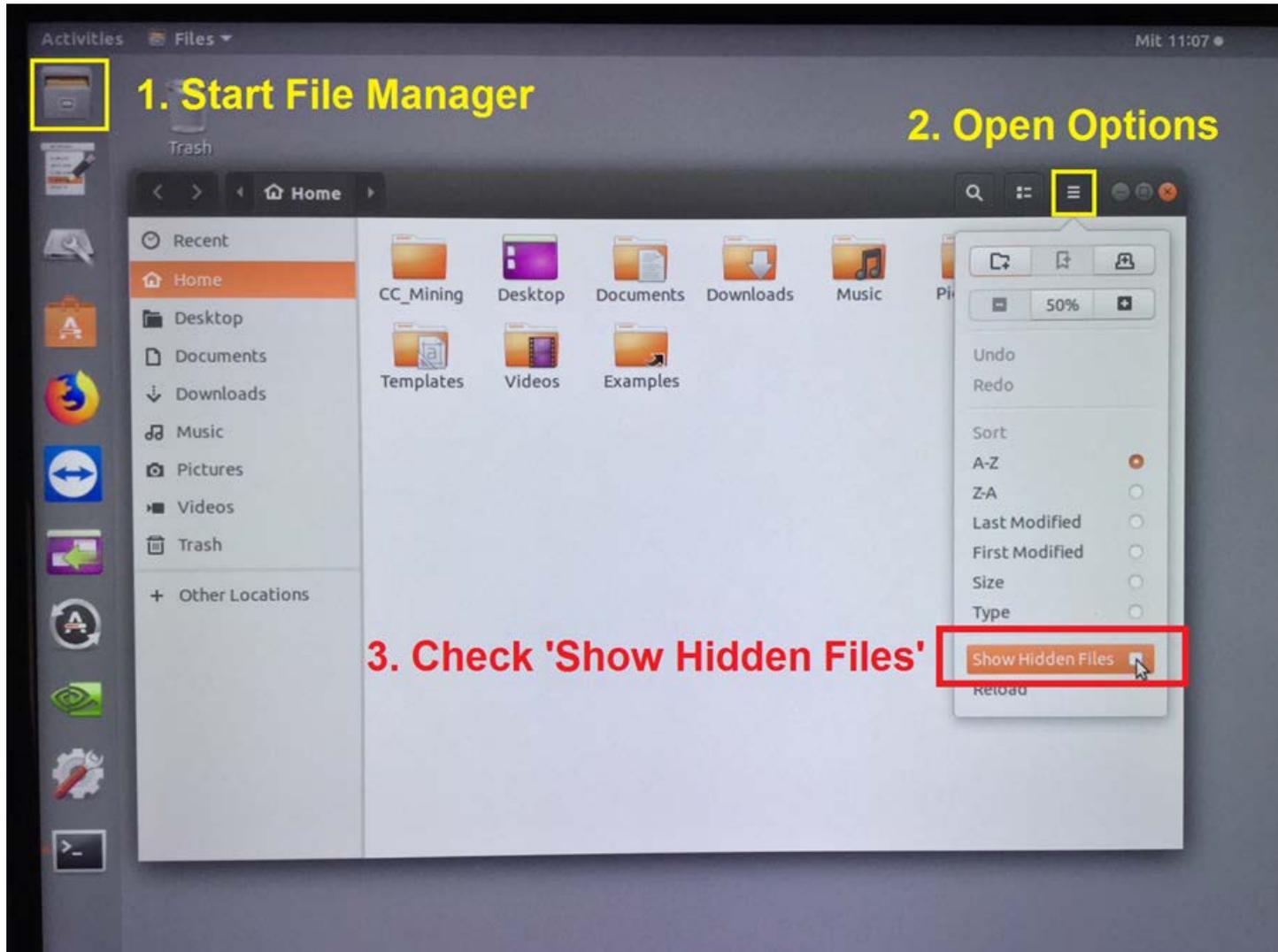
[daemon]
AutomaticLoginEnable=True
AutomaticLogin=swisstrader
# Uncomment the line below to force the login screen to use Xorg
#WaylandEnable=false
# Enabling automatic login

# Enabling timed login
# TimedLoginEnable = true
# TimedLogin = user1
# TimedLoginDelay = 10

[security]
```

A red box highlights the change from `#WaylandEnable=false` to `WaylandEnable=false`. An orange arrow points from the original line to the modified line. A green text overlay on the left reads: "Xorg works after adaption and the AutoLogin too!".

20. Activate option to **show hidden files and folders** in the Files Manager:



21. **Changing the SSH port #** from default 'Port 22' to a port number of your choice  $\leq 65536$  by running in a new opened terminal with:

```
sudo nano /etc/ssh/sshd_config
```

and edit the file with a new port #. **To control the mining rig by SSH over the internet each host (mining rig) needs its own unique port number!**  
Enter CTRL+X, then 'Y' and press key 'ENTER' to save the changed config file.

**OR** by the longer way as follows:

Open a terminal and go to 'root' area by running twice the command line `cd ..` and check with command line `//` the directory content.  
Go to subdirectory `etc` by `cd etc` and here to subdirectory `ssh` by `cd ssh`. Check again the directory content with command line `//`.

**OR**

Open a terminal and go to 'root' area by running the command line `cd /etc/ssh` and check with command line `//` the directory content.  
Now call the file `sshd_config` to justify with the text editor `nano` by `sudo nano sshd_config`. Move cursor up to the line '# Port 22' and change the line to 'Port xxxxx'. The integer number xxxxx is the port number of your choice, but not higher than number 65536.

Enter CTRL+X, then 'Y' and press key 'ENTER'. The port number of SSH service is now justified.

Finally check the it by using command line `sudo nano sshd_config`.

Check the SSH functionality:

```
START CONNECTION VIA SSH:  $ ssh <username>@<device name> -p <port number>
                           $ ssh <user_01>@nuc7i3bnh-1 -p 12345                // local network connection
                           $ ssh <user_02>@nuc7i3bnh-2 -p 65432                // local network connection
                           $ ssh <ip address>@<device name> -p <port number>
                           $ ssh 123.456.789.321 -p 43210                    // Internet connection
                           $ ssh your_user_name@123.456.789.321 : 43210      // Internet connection
STOP CONNECTION VIA SSH:  $ exit and press ENTER (this changes the SSH status to Logout, Connection to nuc7i3bnh-1 is closed.)
```

22. Open 'Ubuntu Software' by clicking on the orange icon in the task bar and search for 'Alacarte'. Click on the icon and install 'Main Menu'.  
Launch the application after installation and add it to the Favorites by right-click on Main-Menu icon in the left favorite task bar to fix it at favorites.

23. Open 'Ubuntu Software' by clicking on the orange icon in the task bar and search for 'x11'. Install the application 'X11VNC Server'.

24. After the installation is finished duplicate the bootable SSD by running the command line:

```
$ sudo dcfldd if=/dev/sda of=/dev/sdb bs=1M conv=noerror,sync // this clones block by block identically the disk
```

OR

use the 'Clone Disk' feature of **Acronis True Image Home 2014** ( ATIH 2014 ).

+++++

## Some helpful support and few examples for beginners in using Linux:

25. For running the **iftop** or **iotop** take command line **sudo iftop** and **sudo iotop** in separate terminals.
26. Standard installation in Ubuntu is help program **top**. To start the program run in a terminal the simple command **top**.
27. The program **htop** runs simple with command **htop**.
28. Running **df** or **man df** for user manual of df (df reports file system disk space usage). Running **free -h** or **free --help** displays the amount of free and used memory in the system.
29. Running **stress --cpu 4** shows the usage of 4 cores of a CPU.
30. **CTRL + C** exits a running programm in a terminal, and the command **'exit'** in a terminal executed with 'ENTER' will close this terminal.
31. Running the command line **'apropos copy'** opens the subcommands of command **copy**.
32. Double execution in one command line works as follows: `$ sudo chattr -i /etc/X11/xorg.conf && sudo chmod 644 /etc/X11/xorg.conf`
33. To check the programs run **\$ man 'xxx'** in a terminal. For example **'\$ man watch'** opens user manual of program **watch**.
34. Running **[command or program name] --help** or **[command or program name] -h** shows the subcommands of a program.
35. Running **history -c -w** to clear history entries and overwrite immediately the clearance of history.
36. If you have to install a **\*.run** file, then run as an example here **\$ sudo sh ./nvidia\_driver\_v390xxx.run**
37. In the prior line 36 the part **'./'** means, the file is in the same directory of the shown path in the terminal.
38. The part **'~/'** substitutes the path part of **/home/<user>/**.

+++++

## Prepare the MiningRig for mining of Crypto Currencies:

39. Open File Manager and create inside of HOME folder a new folder named '**CryptoCoinMining**'
40. Open folder 'CryptoCoinMining' and create the subfolders of your choice '**Bminer**', '**EWBF**' or/and '**ZM**'.
41. Open subfolder 'Bminer' and unpack the file downloaded from <https://bitcointalk.org/index.php?topic=2519271.0>  
Open subfolder 'EWBF' and unpack the file downloaded from <https://bitcointalk.org/index.php?topic=1707546.0>  
Open subfolder '**ZM**' and unpack the file downloaded from <https://bitcointalk.org/index.php?topic=2021765.0>

My preferred mining software is **ZM** developed by DSTM. He always tested his zm development via SSH connection on my built mining rigs with more than 6 GPUs. I trust him and his software.

## Preparing the **nvidia-smi** Power Management Settings at boot with a bash file that starts with root privileges:

42. Create a new file named '**rc.local**' in the folder **/etc/** with command:

```
sudo nano /etc/rc.local
```

Then edit into the file following comands for the bash script:

```
#!/bin/sh -e  
  
## NVIDIA PowerManagement  
/home/<user>/CryptoCoinMining/nvidia-PM.sh || exit 1  
  
exit 0
```

Enter CTRL+X, then 'Y' and press key 'ENTER' to save the bash script.

Now make the file executable with running the command line:

```
sudo chmod 755 /etc/rc.local
```

This file will be always executed at least during the boot process of the machine.

43. Create a new file named 'nvidia-PM.sh' in the folder /home/<user>/CryptoCoinMining/ with command:

```
sudo nano /home/<user>/CryptoCoinMining/nvidia-PM.sh
```

Then edit into the file following comands for the bash script:

```
#!/bin/sh

## NVIDIA Setting for PowerManagement
nvidia-smi -pm 1
nvidia-smi -pl 112

## *****
##
## This bash script will be started with root privileges
## by the file rc.local located in the folder /etc/
##
## *****

#nvidia-smi -pm 1
#nvidia-smi -pl 130                # all devices get the same power limit of 130 W
#nvidia-smi -i 0,1,2,3 -pl 110     # GPU 0, 1, 2 and 3 get power limit 110 W
#nvidia-smi -i 4,5,6,7 -pl 126    # GPU 4, 5, 6 and 7 get power limit 126 W
```

Enter CTRL+X, then 'Y' and press key 'ENTER' to save the bash script.

Now make the file executable with running the command line:

```
sudo chmod 755 /home/<user>/CryptoCoinMining/nvidia-PM.sh
```

Edit this file to adapt the power limit value of your choice.

## Preparing the **nvidia-settings** Over Clocking Settings at boot with a bash file that starts with root privileges:

44. Create a new file named '**nvidia-OC.sh**' in the folder **/home/<user>/CryptoCoinMining/** with command:

```
sudo nano /home/<user>/CryptoCoinMining/nvidia-OC.sh
```

Then edit into the file following comands for the bash script:

```
#!/bin/sh

## NVIDIA Settings for OverClocking
## -----

## setting some terminal variables for easier value change
fanSpeedMonitorGPU="75"
fanSpeed1="70"
fanSpeed2="75"
clockOffsetMonitorGPU="200"
clockOffset="200"
memoryOffsetMonitorGPU="1000"
memoryOffset="1200"

## configure the different OC settings for each GPU
nvidia-settings -a [gpu:0]/GPUFanControlState=1
nvidia-settings -a [fan:0]/GPUTargetFanSpeed=$fanSpeedMonitorGPU
nvidia-settings -a [gpu:0]/GPUPowerMizerMode=1
nvidia-settings -a [gpu:0]/GPUGraphicsClockOffset[3]=$clockOffsetMonitorGPU
nvidia-settings -a [gpu:0]/GPUMemoryTransferRateOffset[3]=$memoryOffsetMonitorGPU

nvidia-settings -a [gpu:1]/GPUFanControlState=1
nvidia-settings -a [fan:1]/GPUTargetFanSpeed=$fanSpeed1
nvidia-settings -a [gpu:1]/GPUPowerMizerMode=1
nvidia-settings -a [gpu:1]/GPUGraphicsClockOffset[3]=$clockOffset
nvidia-settings -a [gpu:1]/GPUMemoryTransferRateOffset[3]=$memoryOffset

nvidia-settings -a [gpu:2]/GPUFanControlState=1
nvidia-settings -a [fan:2]/GPUTargetFanSpeed=$fanSpeed1
nvidia-settings -a [gpu:2]/GPUPowerMizerMode=1
nvidia-settings -a [gpu:2]/GPUGraphicsClockOffset[3]=$clockOffset
nvidia-settings -a [gpu:2]/GPUMemoryTransferRateOffset[3]=$memoryOffset

nvidia-settings -a [gpu:3]/GPUFanControlState=1
nvidia-settings -a [fan:3]/GPUTargetFanSpeed=$fanSpeed1
nvidia-settings -a [gpu:3]/GPUPowerMizerMode=1
nvidia-settings -a [gpu:3]/GPUGraphicsClockOffset[3]=$clockOffset
nvidia-settings -a [gpu:3]/GPUMemoryTransferRateOffset[3]=$memoryOffset
```

```
nvidia-settings -a [gpu:4]/GPUFanControlState=1
nvidia-settings -a [fan:4]/GPUPowerMizerMode=1
nvidia-settings -a [gpu:4]/GPUFanControlState=1
nvidia-settings -a [fan:4]/GPUPowerMizerMode=1
nvidia-settings -a [gpu:4]/GPUGraphicsClockOffset[3]=$clockOffset
nvidia-settings -a [gpu:4]/GPUMemoryTransferRateOffset[3]=$memoryOffset

nvidia-settings -a [gpu:5]/GPUFanControlState=1
nvidia-settings -a [fan:5]/GPUPowerMizerMode=1
nvidia-settings -a [gpu:5]/GPUFanControlState=1
nvidia-settings -a [fan:5]/GPUPowerMizerMode=1
nvidia-settings -a [gpu:5]/GPUGraphicsClockOffset[3]=$clockOffset
nvidia-settings -a [gpu:5]/GPUMemoryTransferRateOffset[3]=$memoryOffset

nvidia-settings -a [gpu:6]/GPUFanControlState=1
nvidia-settings -a [fan:6]/GPUPowerMizerMode=1
nvidia-settings -a [gpu:6]/GPUFanControlState=1
nvidia-settings -a [fan:6]/GPUPowerMizerMode=1
nvidia-settings -a [gpu:6]/GPUGraphicsClockOffset[3]=$clockOffset
nvidia-settings -a [gpu:6]/GPUMemoryTransferRateOffset[3]=$memoryOffset

nvidia-settings -a [gpu:7]/GPUFanControlState=1
nvidia-settings -a [fan:7]/GPUPowerMizerMode=1
nvidia-settings -a [gpu:7]/GPUFanControlState=1
nvidia-settings -a [fan:7]/GPUPowerMizerMode=1
nvidia-settings -a [gpu:7]/GPUGraphicsClockOffset[3]=$clockOffset
nvidia-settings -a [gpu:7]/GPUMemoryTransferRateOffset[3]=$memoryOffset

## Starting the Mining Process
/home/<user>/CryptoCoinMining/StartMiningProcess.sh
```

Enter CTRL+X, then 'Y' and press key 'ENTER' to save the bash script.

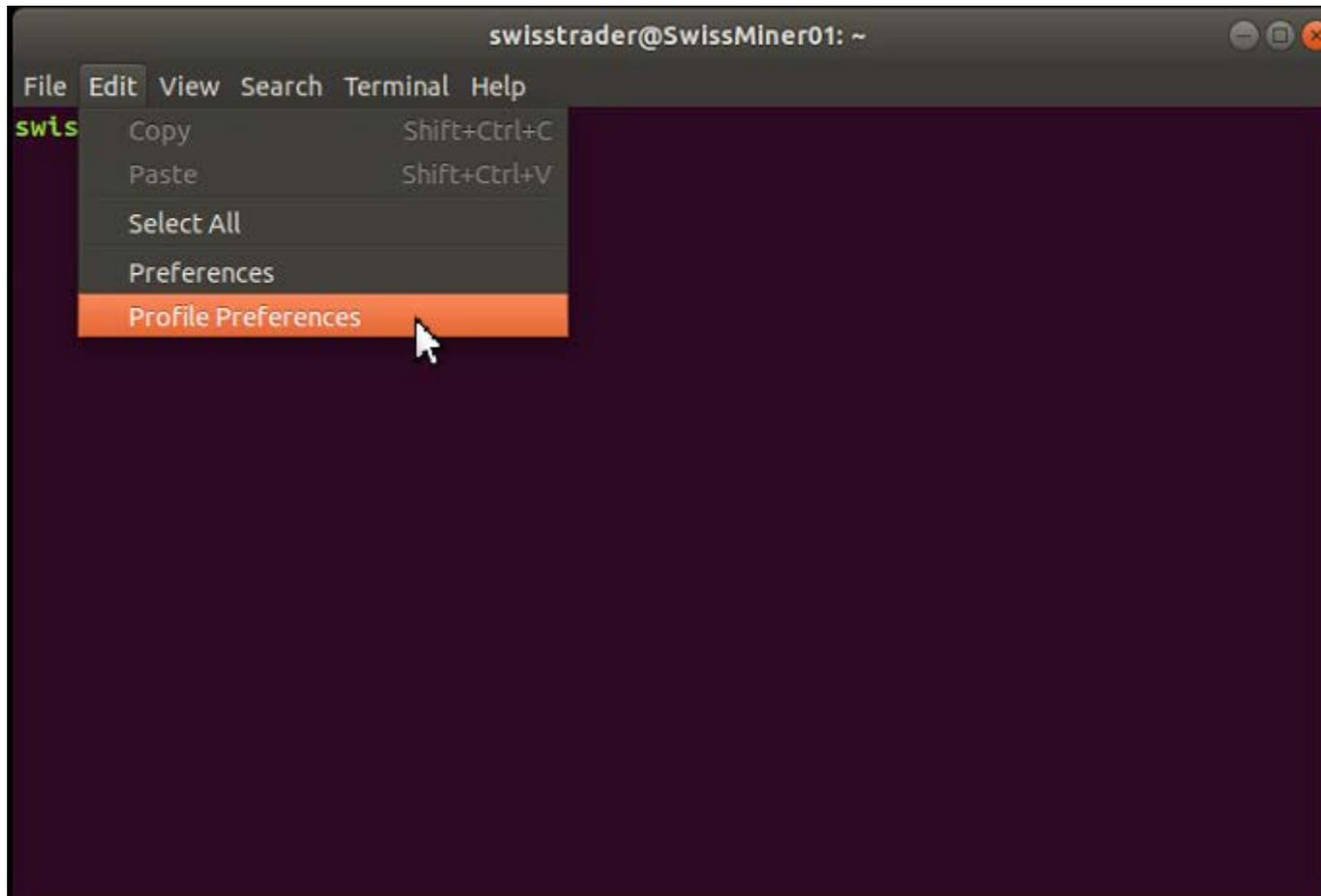
Now make the file executable with running the command line:

```
sudo chmod 755 /home/<user>/CryptoCoinMining/nvidia-OC.sh
```

Edit this file to adapt the values of your choice.

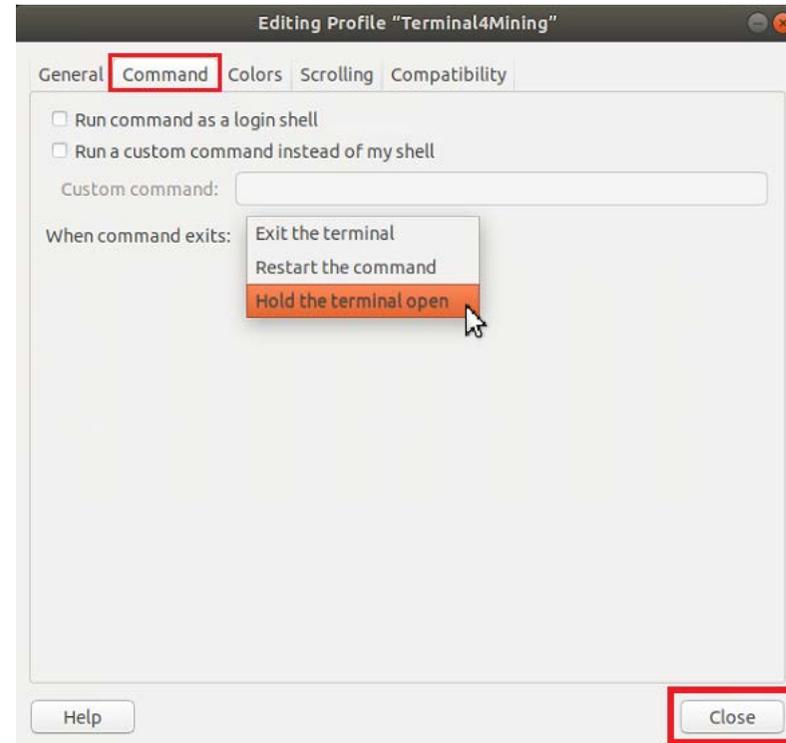
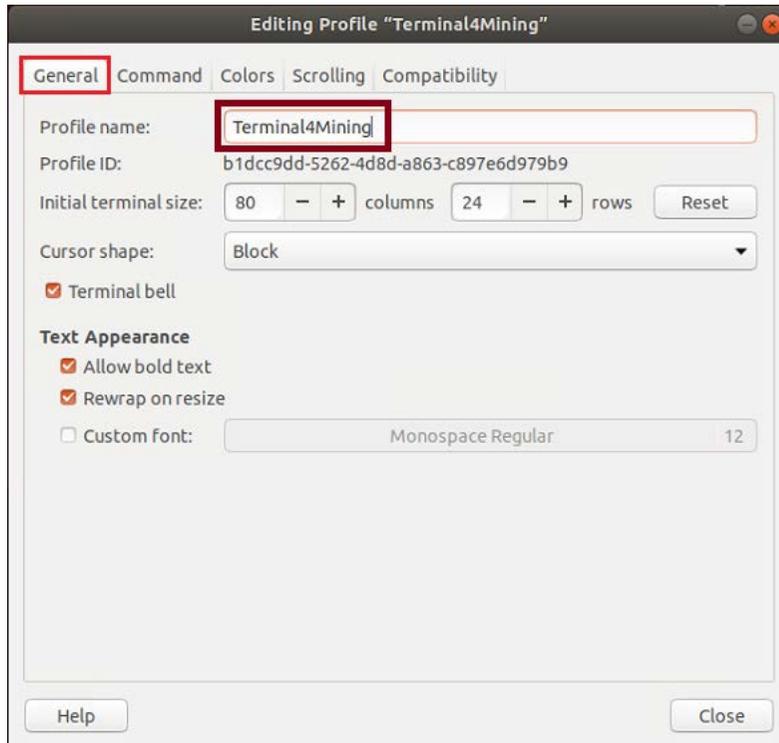
45. Open a new terminal and create in the preferences a new Profile Preference named 'Terminal4Minig'.

Go to Edit => Preferences ...



... and write under tab 'General' into the text field for the Profile name '**Terminal4Mining**'.

Then go to the tab 'Command' and change '**When command exits:**' from default '**Exit the terminal**' to '**Hold the terminal open**'.



Finalize the new Profile by clicking on 'Close' button.

46. Create a new file named 'StartMiningProcess.sh' in the folder /home/<user>/CryptoCoinMining/ with command:

```
sudo nano /home/<user>/CryptoCoinMining/StartMiningProcess.sh
```

Then edit into the file following comands for the bash script:

```
#!/bin/sh

##-geometry WIDTHxHEIGHT(+/-)XOFF(+/-)YOFF (where WIDTH, HEIGHT, XOFF, and YOFF are numbers)
gnome-terminal --window --geometry=80x50+0+0 -e "watch -n 0.25 nvidia-smi" \
  --window-with-profile=Terminal4Mining --geometry=122x50-78+0 \
  -e "/home/<user>/CryptoCoinMining/ZM/Run_ZM.sh"

## script files for different mining software:
## -e "/home/<user>/CryptoCoinMining/Bminer/Run_Bminer.sh"
## -e "/home/<user>/CryptoCoinMining/EWBF/Run_EWBF.sh"
```

Enter CTRL+X, then 'Y' and press key 'ENTER' to save the bash script. Then make the file executable with running the command line:

```
sudo chmod 755 /home/<user>/CryptoCoinMining/StartMiningProcess.sh
```

Edit this file to adapt the mining software of your choice.

47. Create for running the several mining software following bash files:

For ZM:

Note! The config file *BTCZ\_SUPRNOVA\_YourWorkername.cfg* has to exist in the folder of ZM and correctly edited regarding to the manual of ZM!

```
sudo nano /home/<user>/CryptoCoinMining/ZM/Run_ZM.sh
```

```
#!/bin/sh

## Adapt this config file name for changing coin mining
## -----
CONFIGFILE=BTCZ_SUPRNOVA_YourWorkername.cfg

## start to run mining with zm
/home/<user>/CryptoCoinMining/ZM/zm --cfg-file=/home/<user>/CryptoCoinMining/ZM/$CONFIGFILE
```

Enter CTRL+X, then 'Y' and press key 'ENTER' to save the bash script. Then make the file executable with running the command line:

```
sudo chmod 755 /home/<user>/CryptoCoinMining/ZM/Run_ZM.sh
```

uptime: 0 days 00:08:27

host: SwissMiner01

contime: 0 days 00:08:26

server: btcz.suprnova.cc:5586

version: zm 0.6.1

user: swisstrader.Maximus1

| ID    | DEVICE NAME         | °C | FAN % | ∅ SOL/S | ∅ SOL/W | ∅ WATT | SHARE S | LAT |
|-------|---------------------|----|-------|---------|---------|--------|---------|-----|
| 0     | GeForce GTX 1070 Ti | 66 | 75    | 494.15  | 4.46    | 110.77 | 4 / 0   | 35  |
| 1     | GeForce GTX 1070 Ti | 64 | 70    | 511.86  | 4.59    | 111.54 | 16 / 0  | 35  |
| 2     | GeForce GTX 1070 Ti | 63 | 70    | 511.52  | 4.58    | 111.73 | 4 / 0   | 36  |
| 3     | GeForce GTX 1070 Ti | 61 | 70    | 500.42  | 4.48    | 111.63 | 12 / 0  | 36  |
| 4     | GeForce GTX 1070 Ti | 64 | 70    | 510.90  | 4.57    | 111.73 | 16 / 0  | 37  |
| 5     | GeForce GTX 1070 Ti | 58 | 70    | 504.21  | 4.51    | 111.70 | 13 / 0  | 36  |
| 6     | GeForce GTX 1070 Ti | 60 | 70    | 506.22  | 4.53    | 111.75 | 6 / 0   | 36  |
| 7     | GeForce GTX 1070 Ti | 67 | 75    | 514.83  | 4.62    | 111.44 | 12 / 0  | 36  |
| Total |                     | -  | -     | 4054.12 | 4.54    | 892.30 | 83 / 0  | 35  |

For Bminer:

Note! The bash file named *BTCZ\_SUPRNOVA\_YourWorkername.sh* has to exist in the folder of Bminer and correctly edited regarding to the manual of Bminer!

```
sudo nano /home/<user>/CryptoCoinMining/Bminer/Run_Bminer.sh
```

```
#!/bin/sh

## Adapt this bash file name for changing coin mining
## -----
BASHFILE=BTCZ_SUPRNOVA_YourWorkername.sh

## start to run mining with bminer
bash /home/<user>/CryptoCoinMining/Bminer/$BASHFILE
```

Enter CTRL+X, then 'Y' and press key 'ENTER' to save the bash script. Then make the file executable with running the command line:

```
sudo chmod 755 /home/<user>/CryptoCoinMining/Bminer/Run_Bminer.sh
```

The screenshot shows the Bminer web interface with the following data:

| Languages | Uptime    | Speed (Sols / S) | Power (Watts) | Efficiency (Sols / W) | Accepted |
|-----------|-----------|------------------|---------------|-----------------------|----------|
|           | 5 minutes | 4118.15          | 882           | 4.67                  | 28       |

Rejected: 1

### Performance (last 30 minutes)

Two line graphs are shown: Temperature (left) and Speed (right). Both graphs show data for 8 GPUs (GPU 0 to GPU 7) over a 25-second interval from 08:20 to 08:45. The Temperature graph has a y-axis from 0 to 100, and the Speed graph has a y-axis from 0 to 1,000 Sols/S.

### Details

| Device ID | Temperature (°C) | Power (W) | Fan (%) | Speed (Sols / S) | Efficiency (Sols / W) |
|-----------|------------------|-----------|---------|------------------|-----------------------|
| 0         | 65               | 110       | 75      | 509.88           | 4.64                  |
| 1         | 63               | 109       | 70      | 516.51           | 4.74                  |
| 2         | 62               | 111       | 70      | 518.15           | 4.67                  |
| 3         | 61               | 111       | 70      | 503.31           | 4.53                  |
| 4         | 62               | 107       | 70      | 521.43           | 4.87                  |
| 5         | 57               | 109       | 70      | 510.85           | 4.69                  |
| 6         | 59               | 112       | 70      | 516.53           | 4.61                  |
| 7         | 66               | 113       | 75      | 521.49           | 4.61                  |

For EWBF:

Note! The config file named *BTCZ\_SUPRNOVA\_YourWorkername.cfg* has to exist in the folder of EWBF and correctly edited regarding to the manual of EWBF!

```
sudo nano /home/<user>/CryptoCoinMining/EWBF/Run_EWBF.sh
```

```
#!/bin/sh

## Adapt this config file name for changing coin mining
## -----
CONFIGFILE=BTCZ_SUPRNOVA_YourWorkername.cfg

## start to run mining with zm
/home/<user>/CryptoCoinMining/EWBF/miner --config /home/<user>/CryptoCoinMining/EWBF/$CONFIGFILE
```

Enter CTRL+X, then 'Y' and press key 'ENTER' to save the bash script. Then make the file executable with running the command line:

```
sudo chmod 755 /home/<user>/CryptoCoinMining/EWBF/Run_EWBF.sh
```

The screenshot displays the EWBF 0.3.4b mining interface. It is divided into two main sections: 'Statistics of the miner' and 'Statistics of the GPUs'.

**Statistics of the miner**

|                               |           |
|-------------------------------|-----------|
| Miner: 192.168.1.118:2222     | Connected |
| Uptime: 0d 00:01:46           |           |
| Server: btcz.suprnova.cc:5586 | Connected |

**Statistics of the GPUs**

| GPU                 | Temp | Power | Speed      | Efficiency | Accepted | Rejected |
|---------------------|------|-------|------------|------------|----------|----------|
| GeForce GTX 1070 Ti | 62 C | 114 W | 486 Sol/s  | 4.26 Sol/W | 5        | 0        |
| GeForce GTX 1070 Ti | 60 C | 108 W | 493 Sol/s  | 4.56 Sol/W | 1        | 0        |
| GeForce GTX 1070 Ti | 58 C | 113 W | 486 Sol/s  | 4.30 Sol/W | 1        | 0        |
| GeForce GTX 1070 Ti | 61 C | 112 W | 466 Sol/s  | 4.16 Sol/W | 3        | 0        |
| GeForce GTX 1070 Ti | 62 C | 112 W | 491 Sol/s  | 4.38 Sol/W | 0        | 0        |
| GeForce GTX 1070 Ti | 55 C | 111 W | 480 Sol/s  | 4.32 Sol/W | 0        | 0        |
| GeForce GTX 1070 Ti | 60 C | 113 W | 487 Sol/s  | 4.31 Sol/W | 1        | 0        |
| GeForce GTX 1070 Ti | 61 C | 112 W | 494 Sol/s  | 4.41 Sol/W | 4        | 0        |
| Total               | -    | 895 W | 3883 Sol/s | 4.34 Sol/W | 15       | 0        |

EWBF 0.3.4b

48. To control your mining rig via TeamViewer from your Windows PC install PuTTY on your Windows PC and start the connection via PuTTY to the mining rig.

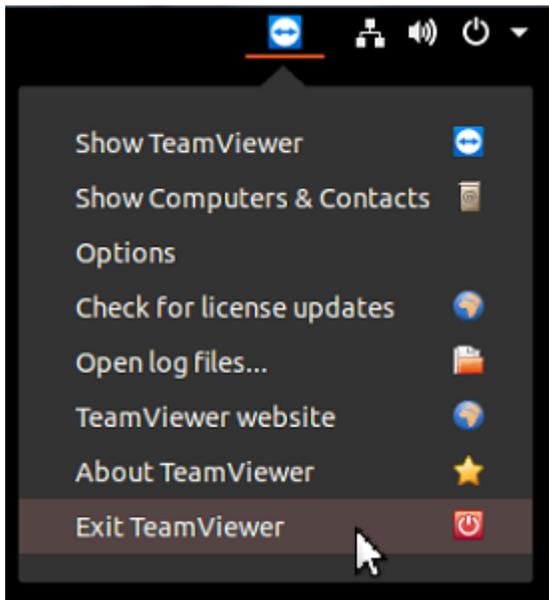
Download link for PuTTY Client for Windows: <https://www.putty.org/>  
<https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>

Login with your username and password in the PuTTY shell and then execute following command line to start TeamViewer on your mining rig:

```
your_user_name@your_device_name:~$ teamviewer -display :0 -auth /var/lib/gdm/:0.Xauth -xkb -forever
```

TeamViewer is now running on the Ubuntu system. Close PuTTY with command 'exit' in the PuTTY shell on your Windows desk, then start TeamViewer on your Windows PC and connect for a Remote Control session to the mining rig.

To close the TeamViewer remote control session exit from the task bar of Ubuntu:



**Please note!** Exit the program TeamViewer saves resources for the mining process.

49. Update to latest NVIDIA driver (recently v396.24):

```
$ sudo add-apt-repository ppa:graphics-drivers/ppa && sudo apt update
```

Then launch “Software & Updates” in the Applications and choose tab named “Additional Driver” utility, here choose the newest driver from the list and apply changes.

**Reboot** the machine.

Open “Software Updater” in the Applications and update the “NVIDIA X server”.

Finally reboot the machine and newest NVIDIA driver will work successfully.

To remove remaining drivers execute following commands:

```
$ dpkg -l | grep -I nvidia
```

```
$ sudo apt-get remove -purge nvidia-... # choose the files from the list to remove it
```

50. Final software service:

```
$ sudo apt-get autoremove -y
```

```
$ sudo apt-get autoclean -y
```

51. Notes: