

ABC Multi GPU Mining: The easy way

A tutorial by Liam Stojanovic

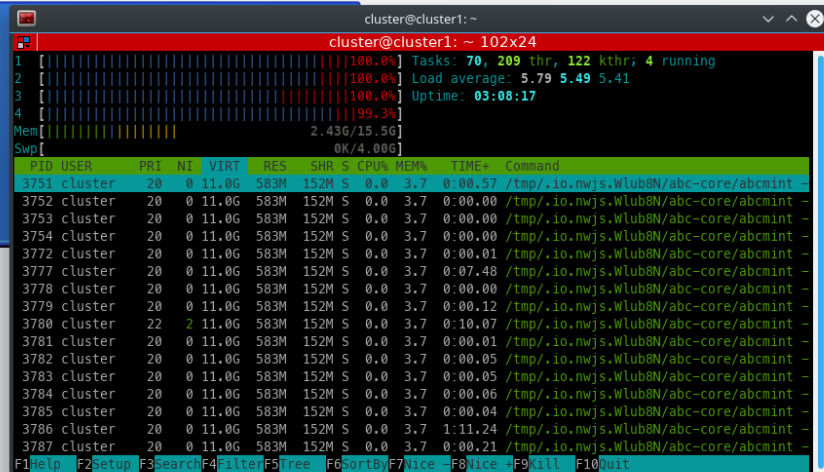
<https://www.linkedin.com/in/liam-stojanovic/>

Overview

- This tutorial will enable you to...
 - Mine ABC using multiple GPUs
 - This method also works for single-card computers
 - Easily manage your mining rig
 - No more plugging in a keyboard, mouse, and monitor to your rig
 - Perform remote administration securely
 - View stats, update the system, and more from anywhere in the world
 - This guide assumes you are using Ubuntu for your personal computer as well (in other words, the computer you'll be using to access your rig remotely)
 - **Commands in BLUE are run on your client PC**, while **commands in RED are to be run on the mining rig**.

Hardware

- The mining rig I am using has...
 - 4x Nvidia 1080Ti GPUs
 - Intel Pentium CPU G4560 @ 3.50GHz
 - 2 cores, 4 threads. Allows us to mine with 4 cards max.
 - Each GPU seems to require 1 thread for mining. Keep this in mind when choosing hardware. My configuration would not work if I had more than 4 GPUs.
 - The top right image shows all four cards mining on the pentium, with the system resource usage
 - 120GB SSD
 - 16GB RAM (you don't need this much, 4GB should be fine).
- You'll also need a USB drive with Ubuntu Server 18.04 loaded onto it.



The screenshot shows a terminal window titled 'cluster@cluster1: ~ 102x24'. The top section displays system statistics: 'Tasks: 70, 209 thr, 122 kthr; 4 running', 'Load average: 5.79 5.49 5.41', and 'Uptime: 03:08:17'. Below this, memory usage is shown as 'Mem[2.43G/15.5G]' and swap as 'Swp[0K/4.00G]'. The main part of the terminal is a table of running processes, with columns for PID, USER, PRI, NI, VIRT, RES, SHR, S, CPU%, MEM%, TIME+, and Command. The table lists several instances of the 'cluster' user running the 'abcmin' command, with varying resource usage. At the bottom, a navigation bar shows function keys: F1 help, F2 Setup, F3 Search, F4 Filter, F5 Tree, F6 SortBy, F7 View, F8 View, F9 Kill, and F10 Quit.

```
cluster@cluster1: ~ 102x24
1 [|||||] 100.0% Tasks: 70, 209 thr, 122 kthr; 4 running
2 [|||||] 100.0% Load average: 5.79 5.49 5.41
3 [|||||] 100.0% Uptime: 03:08:17
4 [|||||] 99.3%
Mem[|||||] 2.43G/15.5G
Swp[|||||] 0K/4.00G
PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
3751 cluster 20 0 11.0G 583M 152M S 0.0 3.7 0:00.57 /tmp/.io.nwjs.Wlub8N/abc-core/abcmin
3752 cluster 20 0 11.0G 583M 152M S 0.0 3.7 0:00.00 /tmp/.io.nwjs.Wlub8N/abc-core/abcmin
3753 cluster 20 0 11.0G 583M 152M S 0.0 3.7 0:00.00 /tmp/.io.nwjs.Wlub8N/abc-core/abcmin
3754 cluster 20 0 11.0G 583M 152M S 0.0 3.7 0:00.00 /tmp/.io.nwjs.Wlub8N/abc-core/abcmin
3772 cluster 20 0 11.0G 583M 152M S 0.0 3.7 0:00.01 /tmp/.io.nwjs.Wlub8N/abc-core/abcmin
3777 cluster 20 0 11.0G 583M 152M S 0.0 3.7 0:07.48 /tmp/.io.nwjs.Wlub8N/abc-core/abcmin
3778 cluster 20 0 11.0G 583M 152M S 0.0 3.7 0:00.00 /tmp/.io.nwjs.Wlub8N/abc-core/abcmin
3779 cluster 20 0 11.0G 583M 152M S 0.0 3.7 0:00.12 /tmp/.io.nwjs.Wlub8N/abc-core/abcmin
3780 cluster 22 2 11.0G 583M 152M S 0.0 3.7 0:10.07 /tmp/.io.nwjs.Wlub8N/abc-core/abcmin
3781 cluster 20 0 11.0G 583M 152M S 0.0 3.7 0:00.01 /tmp/.io.nwjs.Wlub8N/abc-core/abcmin
3782 cluster 20 0 11.0G 583M 152M S 0.0 3.7 0:00.05 /tmp/.io.nwjs.Wlub8N/abc-core/abcmin
3783 cluster 20 0 11.0G 583M 152M S 0.0 3.7 0:00.05 /tmp/.io.nwjs.Wlub8N/abc-core/abcmin
3784 cluster 20 0 11.0G 583M 152M S 0.0 3.7 0:00.06 /tmp/.io.nwjs.Wlub8N/abc-core/abcmin
3785 cluster 20 0 11.0G 583M 152M S 0.0 3.7 0:00.04 /tmp/.io.nwjs.Wlub8N/abc-core/abcmin
3786 cluster 20 0 11.0G 583M 152M S 0.0 3.7 1:11.24 /tmp/.io.nwjs.Wlub8N/abc-core/abcmin
3787 cluster 20 0 11.0G 583M 152M S 0.0 3.7 0:00.21 /tmp/.io.nwjs.Wlub8N/abc-core/abcmin
F1 help F2 Setup F3 Search F4 Filter F5 Tree F6 SortBy F7 View F8 View F9 Kill F10 Quit
```

Software

- Ubuntu Server 18.04 as Operating System for miner
 - **OpenSSH, tightvncserver, ngrok**
 - remote administration
 - **XFCE**
 - desktop environment
 - **ABC-mint.com Second Foundation**
 - Mining software
 - Requires CUDA 10.2 or later

Step 1 – Install Ubuntu Server

- Install Ubuntu Server on your machine, enable OpenSSH
- Once install and reboot is complete:
 - Verify SSH works.

```
$ ssh abcminer@YOUR-IP-HERE
```

```
$ sudo apt update
```

```
$ sudo apt upgrade
```

- Update the system packages

Step 2 – Security (optional, but recommended)

- Rather than authenticating with a password, we will switch to authenticating with **RSA key pairs**.

```
$ ssh-keygen -b 4096
```

- Creates RSA keypair

```
$ ssh-copy-id -i .ssh/your_key abcminer@YOUR-IP-HERE
```

- Copies key to the mining rig

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

- Scroll to PasswordAuthentication, change to PasswordAuthentication no

```
$ sudo systemctl restart ssh
```

- Verify that you can still SSH into the machine

```
$ ssh -i .ssh/your_key abcminer@YOUR-IP-HERE
```

Step 2 – Security, continued...

- **Setting up UFW (firewall)**

- `$ sudo ufw allow OpenSSH`

- Allows connections to port 22

- `$ sudo ufw enable`

- Enable the firewall

- `$ sudo ufw status`

- Shows that SSH connections are still allowed.

- At this point, it's recommended that you disconnect and reconnect your SSH session to verify your firewall works as intended.

Step 3 – Prerequisite Software

- On your **client machine**, install a VNC client.
 - For Ubuntu, I recommend RealVNC (<https://www.realvnc.com/en/connect/download/viewer/linux/>)
- Now SSH onto your **mining rig** in order to install the Desktop Environment.

\$ sudo apt install xfce4 xfce4-goodies

- Installs the desktop environment

\$ sudo apt install tightvncserver

- Installs VNC server

\$ vncserver

- Starts the VNC server. This command prompts you for a password the first time you run it. The password doesn't need to be long or complicated. You don't need a view-only password, either.

\$ vncserver -kill :1

- Stop VNC in order to edit configuration files.

Step 3 – Prerequisite Software, continued...

```
$ cp ~/.vnc/xstartup ~/.vnc/xstartup.bak
```

- Back up the xstartup file before modifying it.

```
$ nano ~/.vnc/xstartup
```

- Edit the first three lines of the file to match the following:

```
#!/bin/bash
```

```
xrdb $HOME/.Xresources
```

```
startxfce4 &
```

- Save the file and close nano

```
$ sudo chmod +x ~/.vnc/xstartup
```

- Mark the file as executable

```
$ vncserver
```

- Starts the VNC server, now with the new configuration.

Step 3 – Prerequisite Software, Continued...

- Install CUDA

```
$ wget https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64/cuda-ubuntu1804.pin
```

```
$ sudo mv cuda-ubuntu1804.pin /etc/apt/preferences.d/cuda-repository-pin-600
```

```
$ wget http://developer.download.nvidia.com/compute/cuda/10.2/Prod/local_installers/cuda-repo-ubuntu1804-10-2-local-10.2.89-440.33.01_1.0-1_amd64.deb
```

```
$ sudo dpkg -i cuda-repo-ubuntu1804-10-2-local-10.2.89-440.33.01_1.0-1_amd64.deb
```

```
$ sudo apt-key add /var/cuda-repo-10-2-local-10.2.89-440.33.01/7fa2af80.pub
```

```
$ sudo apt update
```

```
$ sudo apt -y install cuda
```

- See Nvidia's website for more information (https://developer.nvidia.com/cuda-downloads?target_os=Linux&target_arch=x86_64&target_distro=Ubuntu&target_version=1804&target_type=deblocal)
- **Reboot the system after CUDA install is complete**

Step 3 – Prerequisite Software, continued...

- Download the ABCMint second foundation software onto your personal computer.

```
$ scp -i .ssh/your_key ABCMint-Second-Foundation.tar.gz abcminer@YOUR-IP-HERE:/tmp
```

- Copies the ABCMint zip from your machine to the miner

```
$ cp /tmp/ABCMint-Second-Foundation.tar.gz ~/
```

```
$ tar xvf ABCMint-Second-Foundation.tar.gz
```

```
$ cd ABCMint-Second-Foundation/
```

```
$ chmod +x install.sh
```

```
$ ./install.sh
```

- All the installer script seems to do is create a desktop link to the executable.

- That's all the prerequisite software we need. Now, we can connect to our miner's desktop using SSH and your VNC client.

Step 4 – Starting the miner

- Connect to your miner using the following syntax:

```
$ ssh -i .ssh/your_key -L 5901:localhost:5901 abcminer@YOUR-IP-HERE
```

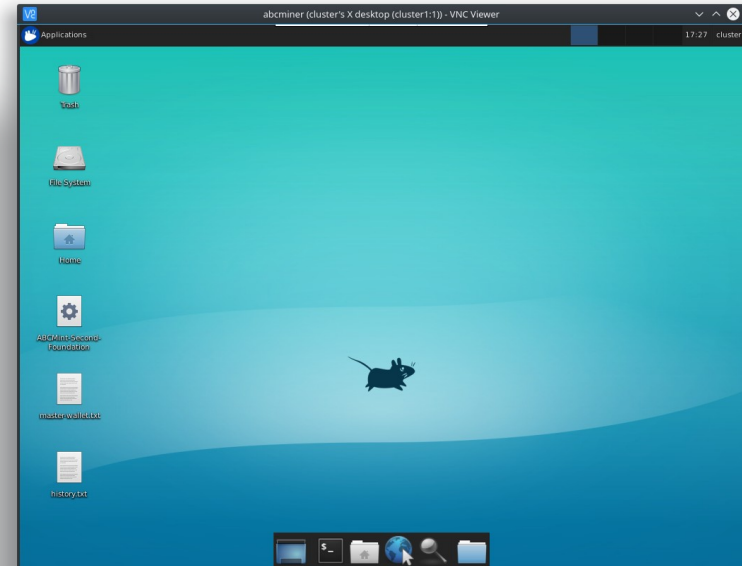
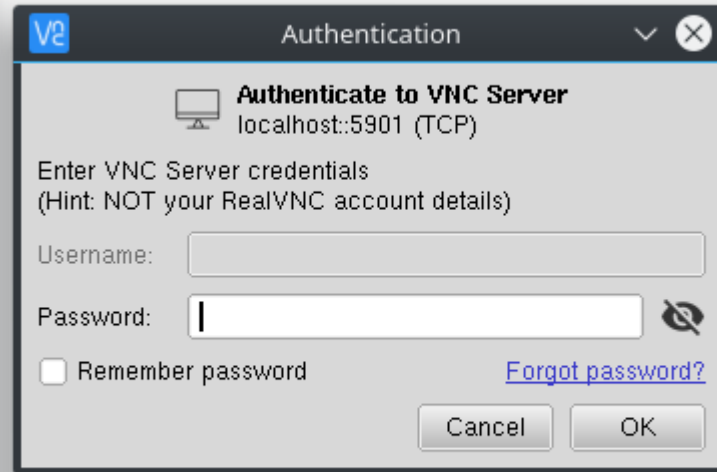
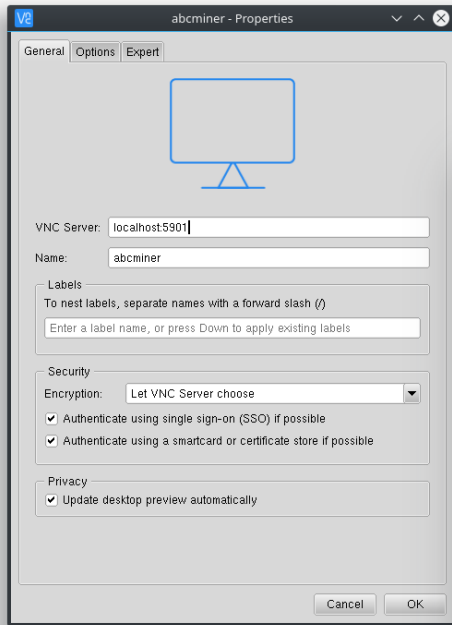
- The “-L” argument will forward any arbitrary data received to the local port 5901 to the remote machine. This allows us to use VNC over SSH.
 - Once the session connects, the port is forwarded.
- Once connected...

```
$ vncserver
```

- It's important to note that with this configuration, you will have to run **vncserver** every time the computer is shutdown or rebooted.

Step 4 – Starting the miner, continued...

- Keep the window with your SSH session open.
- Launch your VNC client, and connect to localhost:5901. See the screenshots below.



Step 4 – Starting the miner, continued...

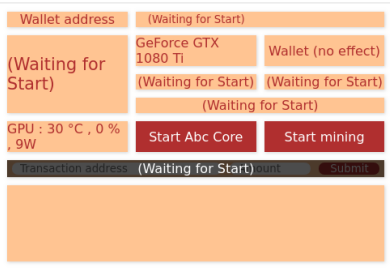
- Click the “ABCMint-Second-Foundation” shortcut on the desktop. The following screen will appear...



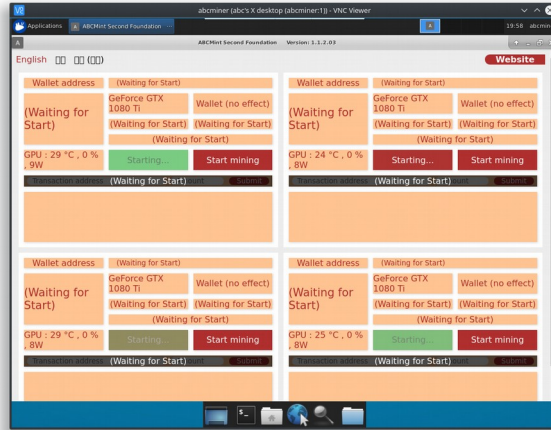
Step 4 – Starting the miner, continued...

- ABCMint Second Foundation software notes
 - For each separate GPU detected on the system, a new panel is generated. Each panel runs its own headless instance of ABCmint.
 - Every time you boot the program, click “Start Abc Core” on each panel, and wait for the blockchain to sync with the network. The first time it will take a while; subsequent startups will take about 2 minutes.
 - Click “Start mining” on each panel once the blockchain is in sync. You are now mining ABC!
 - See the next slide for screenshots

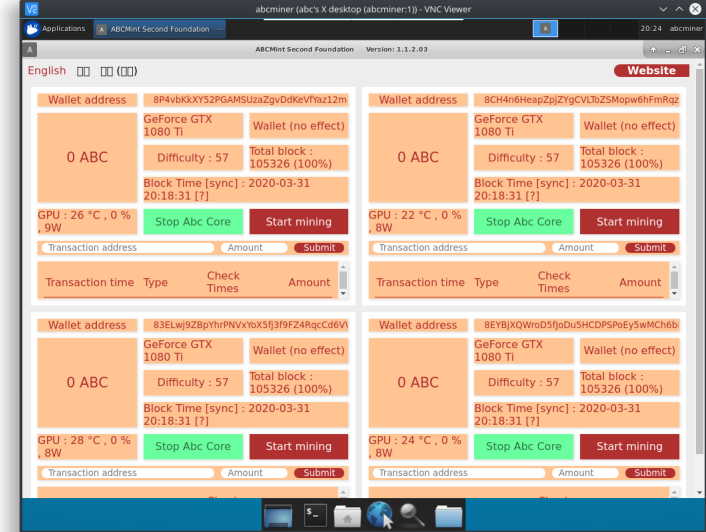
Step 4 – Starting the miner, continued...



Brand new install,
no blockchain sync



Initial blockchain sync for all 4
ABC instances



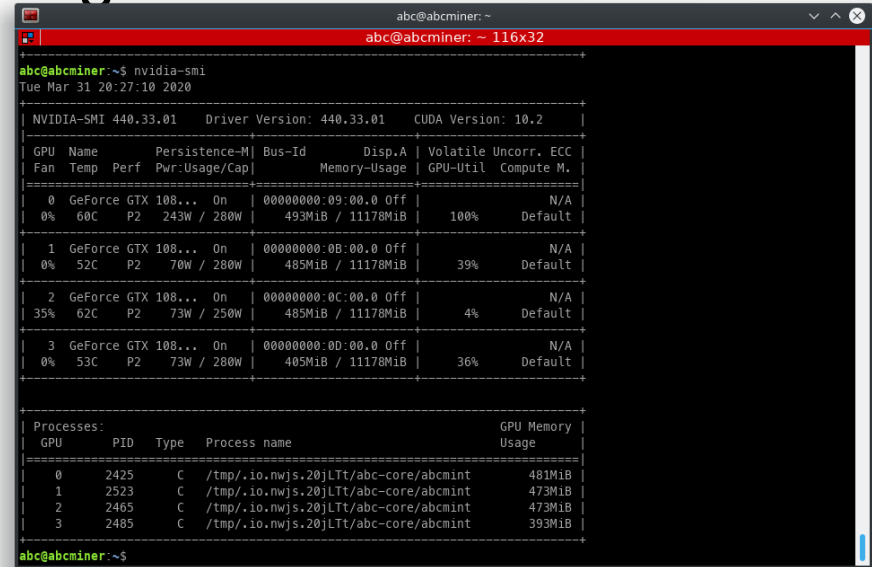
All four ABC instances are synced,
and now ready to mine

Step 4 – Starting the miner, continued...

- Simply click the “Start mining” button for each panel, and you are now successfully mining ABC!



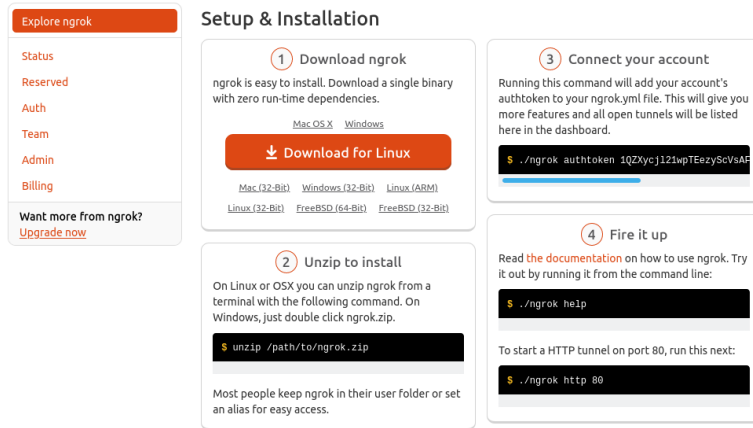
All four cards now mining ABC



Verify miner is running with
\$ nvidia-smi

Step 5 – Remote administration

- The mining rig is now fully functional, and can be administrated while connected to the same LAN
- **Ngrok** allows us to administrate the system from any location, with minimal setup
 - To start, create an account at <https://ngrok.com/>. The following screen will appear once logged in



The screenshot displays the Ngrok website's 'Setup & Installation' page. On the left is a navigation menu with links: Status, Reserved, Auth, Team, Admin, Billing, and a 'Want more from ngrok? Upgrade now' button. The main content area is titled 'Setup & Installation' and contains four numbered steps:

- 1 Download ngrok**: ngrok is easy to install. Download a single binary with zero run-time dependencies. It provides download links for Mac OS X, Windows, and Linux (ARM). A prominent orange button says 'Download for Linux'. Below this, it lists specific binaries: Mac (32-Bit), Windows (32-Bit), Linux (ARM), Linux (32-Bit), FreeBSD (64-Bit), and FreeBSD (32-Bit).
- 2 Unzip to install**: On Linux or OSX you can unzip ngrok from a terminal with the following command. On Windows, just double click ngrok.zip. A terminal snippet shows: `$ unzip /path/to/ngrok.zip`. A note at the bottom says: 'Most people keep ngrok in their user folder or set an alias for easy access.'
- 3 Connect your account**: Running this command will add your account's auth token to your ngrok.yml file. This will give you more features and all open tunnels will be listed here in the dashboard. A terminal snippet shows: `$./ngrok authtoken 1QZkycj121wpTEezyScVsAF`.
- 4 Fire it up**: Read the documentation on how to use ngrok. Try it out by running it from the command line: A terminal snippet shows: `$./ngrok help`. Below this, it says: 'To start a HTTP tunnel on port 80, run this next:' and a terminal snippet shows: `$./ngrok http 80`.

Step 5 – Remote administration, continued...

- Copy the download location from the “Download for Linux” button.
- SSH onto your mining rig

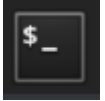
```
$ wget -P ~/ https://...../ngrok-stable-linux-amd64.zip
```

```
$ unzip ngrok-stable-linux-amd64.zip
```

```
$ ./ngrok authtoken YOUR-AUTH-TOKEN-HERE
```

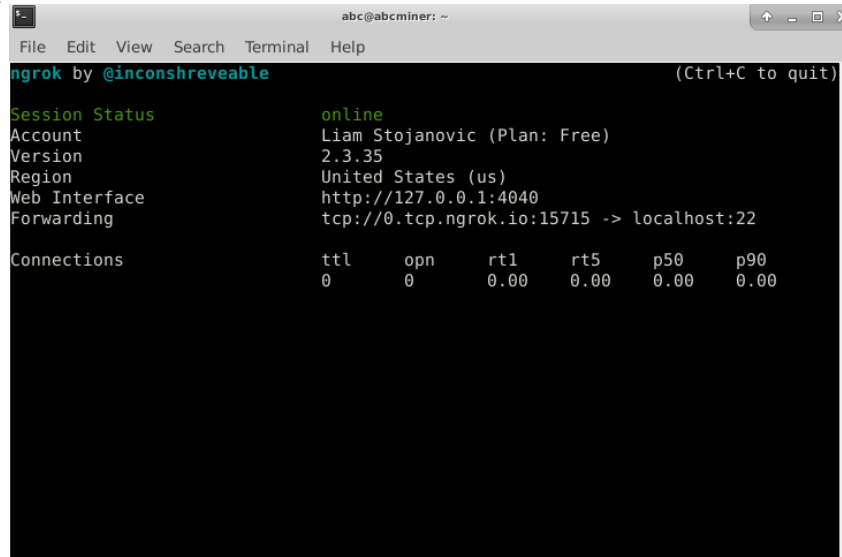
Step 5 – Remote administration, continued...

- Open your VNC client, and open the terminal emulator



`$./ngrok tcp 22`

- An output similar (but *not* identical) will appear in the window



```
abc@abcminer: ~  
File Edit View Search Terminal Help  
ngrok by @inconshreveable (Ctrl+C to quit)  
  
Session Status      online  
Account             Liam Stojanovic (Plan: Free)  
Version             2.3.35  
Region              United States (us)  
Web Interface       http://127.0.0.1:4040  
Forwarding           tcp://0.tcp.ngrok.io:15715 -> localhost:22  
  
Connections          ttl    opn    rt1    rt5    p50    p90  
0                  0      0.00   0.00   0.00   0.00
```

The “forwarding” line contains information we need to connect.

Our internet-accessible IP is 0.tcp.ngrok.io, and the port number is 15715.

This info can also be found on <https://dashboard.ngrok.com/status>

As long as you have an account and registered ngrok using your authtoken, your port number should stay the same until you disconnect.

Step 5 – Remote administration, continued...

- Using the last slide as an example, all we have to do is change a few parameters when connecting to the computer via SSH.
- Our IP is 0.tcp.ngrok.io, our port is 15715

```
$ ssh -i your_key -L 5901:localhost:5901 abc@0.tcp.ngrok.io -p 15715
```

The usual “The authenticity of host '[0.tcp.ngrok.io]:15715 ([18.188.14.65]:15715)' can't be established.” message will appear. Type “yes”.

```
abc@abcminer:~$
```

- That's it! You've successfully set up remote administration. Run the above command anywhere, and you can connect to your mining rig and view the desktop!

Recap

- When connecting to your mining rig in the future...

- **LAN**

- ```
$ ssh -i ~/path/to/your/key -L 5901:localhost:5901 username@IP
```

- **Remote Administration through Ngrok**

- ```
$ ssh -i ~/path/to/your/key -L 5901:localhost:5901 username@IP -p  
port
```