V.2

TUTORIAL

HOW TO RUN A CREDITS NODE

ON A PC OR ON A LINUX VPS (WITH PC OR MAC)

Credits

@Toki1975 : technical issues, layout (source file)
@Micmac01: technical issues, technical advisor
@Will_run : technical issues, technical advisor
@HaryBeno: technical issues, digital publishing (ePub, Mobi, PDF)

From **French Credits Telegram channel**

DISCLAIMER:
This document isn't an official Credits publication.

For any technical questions,
please join Credits Technical Telegram channel:

t.me/creditstechnical

Useful links :

Credits official website : credits.com

Credits Main Telegram channel : t.me/creditscom

Buy Credits (CS) : kucoin.com/#/trade/CS-ETH

Credits Testnet: developers.credits.com/en/Articles/a_TestNet

Credits Mainnet: developers.credits.com/en/Articles/a_MainNet

Business and Partnerships: hb@credits.com

TABLE OF CONTENTS

INTRODUCTION

This guide aims to help Credits community members who wants to run a node on their PC or on a Linux VPS, and to guide them in their first steps.

MAIN PURPOSE OF THIS TUTORIAL

The purpose of this tutorial is to allow Windows users to run a Credits node with ease, and to teach to Windows and Mac OS users with no Linux knowledge how to install, configure and manage a CS Node installed on a Linux VPS (without Graphical interface), from a Windows or Mac OS operating system.

You don't need to have an IT background to run this tutorial. We will always choose the simple and easiest way to do things. And most of the Linux commands that you will find here can be copy-pasted from this document to the Linux command line.

WHAT YOU WILL BE ABLE TO DO AT THE END OF THE TUTORIAL

♦ **Configuring a PC in order to run a node, or a rented Linux VPS** under Ubuntu 18.04 (without graphical interface)

♦ **Running a Credits node** on your PC or your VPS (on Credits Testnet or Mainnet)

♦ **Managing your VPS** through the Tmux software and even closed, it will continue to run 24/7

WHAT IS A CREDITS NODE?

A node is a client-side application that is installed on the user equipment. The node processes and stores transactions, executes and confirms smart contract rules requests processing from third-party systems and provides data when requested.

WHAT ARE ITS FEATURES?

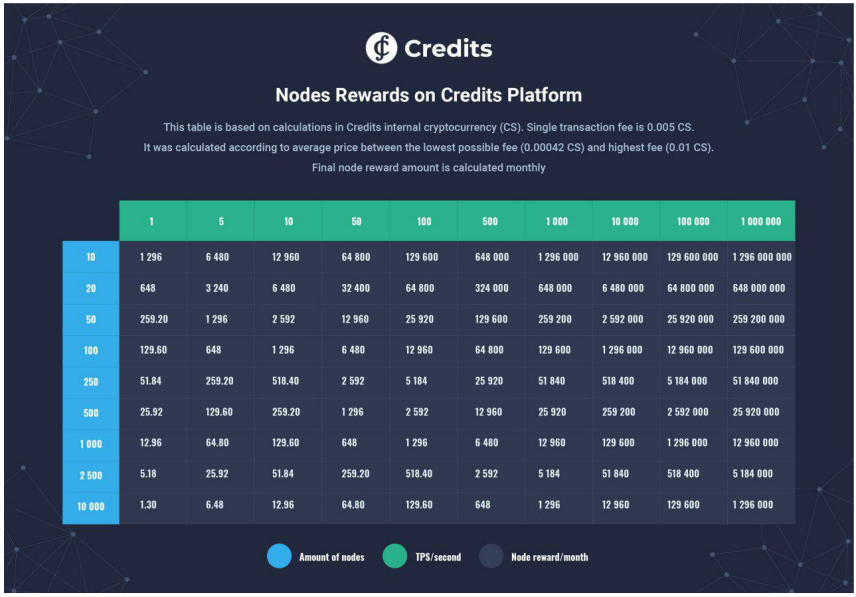
Implemented in C++, It's a complete node and digital wallet that provides the following set of functions:

- ♦ Participation in the consensus algorithm
- ♦ Storage of public registry blockchain with all transactions
- ♦ Performing value transfer transactions: CS and tokens released on the Credits platform
- ♦ Allowing account balance check
- ♦ Creating an account/digital wallet
- ♦ Developing, compilation and deploying of a smart contract
- ♦ Issuance of internal tokens that work with the platform/Credits protocol
- ♦ Smart contract testing
- ♦ Execution of smart contract functions/methods (like 'execute', 'save', 'approve', 'view', 'check', etc.)

WHAT ARE THE BENEFITS OF RUNNING A CREDITS NODE?

The first benefit is quite symbolic, it helps the Credits blockchain network for consensus decisions, transactions processing and storage of datas on Credits blockchain.

The second benefit for a user who runs a Credits node is that he get rewards for each transaction validation he participates in. You will find below the detailed table of Credits node rewards.



	1	5	10	50	100	500	1 000	10 000	100 000	1 000 000
10	1 296	6 480	12 960	64 800	129 600	648 000	1 296 000	12 960 000	129 600 000	1 296 000 000
20	648	3 240	6 480	32 400	64 800	324 000	648 000	6 480 000	64 800 000	648 000 000
50	259.20	1 296	2 592	12 960	25 920	129 600	259 200	2 592 000	25 920 000	259 200 000
100	129.60	648	1 296	6 480	12 960	64 800	129 600	1 296 000	12 960 000	129 600 000
250	51.84	259.20	518.40	2 592	5 184	25 920	51 840	518 400	5 184 000	51 840 000
500	25.92	129.60	259.20	1 296	2 592	12 960	25 920	259 200	2 592 000	25 920 000
1 000	12.96	64.80	129.60	648	1 296	6 480	12 960	129 600	1 296 000	12 960 000
2 500	5.18	25.92	51.84	259.20	518.40	2 592	5 184	51 840	518 400	5 184 000
10 000	1.30	6.48	12.96	64.80	129.60	648	1 296	12 960	129 600	1 296 000

© El Patron (TG: @Pablo_Emilio_Gaviria)

REQUIRED ELEMENTS

♦ **A Personal Computer** under Windows or a Mac (only with VPS)

♦ **Version 4.2 (or newer) of Credits Testnet** or **Mainnet**, with a **stake of more than 50.000 CS** in node wallet

♦ **JRE (for PC node)** or **Java client for Linux (for VPS node)**

♦ **A rented Linux VPS** (tutorial made on Ubuntu 18.04)

♦ **PuTTY and FileZilla** softwares (freewares and available for both Windows and Mac)

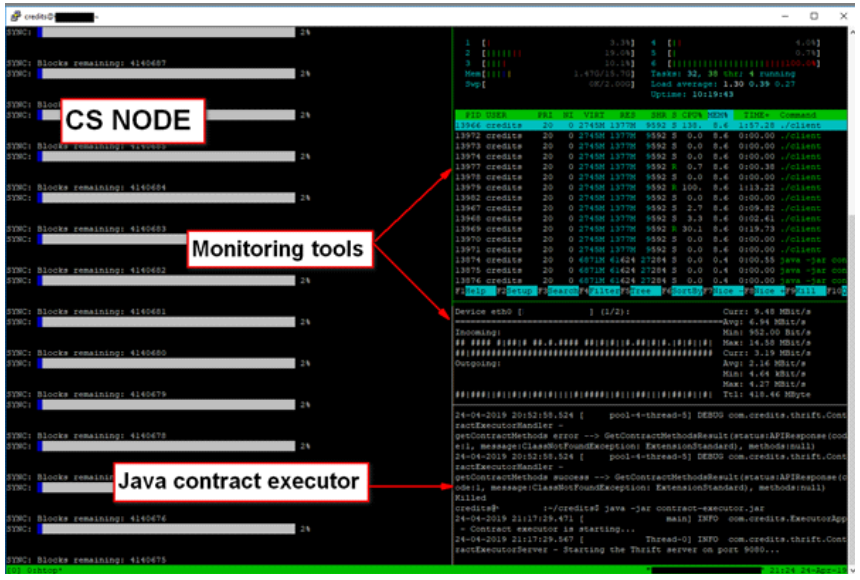
♦ **Few Linux monitoring tools** installed on the VPS

FINAL RESULT

Once you will have set your Credits node (either on your PC or on your Linux VPS), your node will work and you will be a part of Credits blockchain.

The following screenshot shows the final result of this tutorial with a Linux VPS. It's a SSH session on the Linux VPS that you will rent and where the Credits Node software will be running.

You can quit it at anytime you want by clicking on the cross, like on any programs you are familiar with under Windows or Mac OS. The Credits Node and other softwares that you have started in this SSH session will continue to run until you connect again on your VPS, or until you stop them manually or reboot your VPS.



SUPPORT OUR WORK!

This tutorial has been made by four french guys passionated by Credits technology, who wished to help Credits community to run a node and earn money in an easy way.

Even if it took us countless hours and days to create it, test it, correct it and publish it, **this 63 pages tutorial is 100% free!**

Nevertheless, if you found it useful and are grateful, feel free to send us a tip, even for a coffee!

No matter the amount, symbolic rewards are always much appreciated, and it encourages us to constantly improve this tutorial, following Credits evolutions.

Enjoy this tutorial and thank you for your support.

BITCOIN (BTC) adress:

39XpiEEtjLMFszEDSkJCZiHBCpZBD8mAks

ETHEREUM (ETH) adress:

0x0364C818dF9c1D7626a9d329a477C53434cAedA8

CREDITS (CS) adress:

Very soon! ;-)

HOW TO RUN A NODE ON YOUR LINUX VPS

STEP 1:
RENTING A LINUX VPS

There are a lot of Linux VPS providers. Choose the one you want, but take a monthly contract so you will not loose money in case you want to stop renting the VPS.

Minimum system requirements are:

- 4 CPU VPS
- 4GB RAM
- 40/50GB disk space
- 3TB bandwidth per month allowed

Once the VPS is rented, you will receive shortly an email from the provider with the credentials and web address of your customer control panel, where you can install/reinstall/reboot/stop/start your VPS. There will be also the IP address of your VPS that you will need to connect on it through SSH (PuTTY) or SFTP (Filezilla).

On your control panel, before starting installation of you Linux VPS, you will be asked to choose a root password and a Linux version (ie: **Ubuntu 18.04**).

Once your VPS is up and running, you can go to **Step 2 (Installation of PuTTY)**.

STEP 2:
INSTALLATION OF PUTTY

PuTTY is a free client SSH (*Secure SHell*) and Telnet (*Terminal Network*) for **Windows and Unix platforms**, along with an Xterm Terminal Emulator which will allow you to connect to your Linux VPS.

For windows users, here is the link to download and the install this free software:

DOWNLOAD

For MacOS users, unfortunately you don't have a .dmg file you can install as easily as the windows version. So, there are two ways to connect on SSH from your MacOS.

1. Follow this YouTube video that explains how to install PuTTY on your Mac in a few minutes:
www.youtube.com/watch?v=rblQ4y9coGg

2. Install one of the proposed SSH clients as alternatives in this article:
beebom.com/putty-for-mac-free-alternative-ssh-clients

STEP 3:
INSTALLATION OF FILEZILLA

FileZilla is a FTP (*File Transfer Protocol*) software that you can use through **Windows and Mac OS X** to upload Credits software node and also download or copy the Public and Private Keys generated when you will have run your CS node for the first time.

In case you want to install a new version of the CS node software, you will be able to delete the folder containing the Credits software and then upload a new version if you don't want to use the Linux command line to delete the Credits folder.

But first thing, you have to download the software and install it.

For Windows (64 bits):

For Mac OS X:

Now that you have a Linux VPS, a SSH client and a SFTP installed on your Windows or Mac, everything is ready to configure your Linux distribution and install your CS Node.

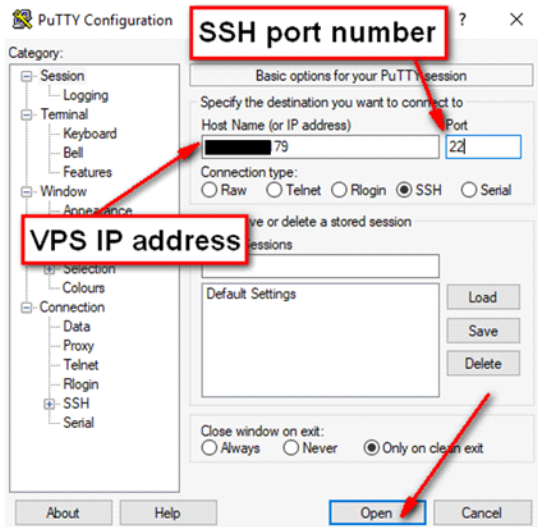
STEP 4:

VPS CONNECTION AND LINUX CONFIGURATION WITH PUTTY (SSH)

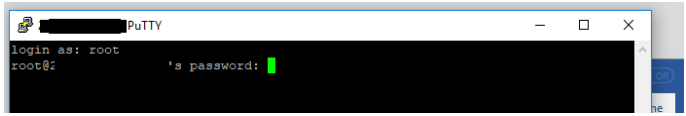
In this part of the tutorial, you have to create a connection with PuTTY software on Linux. Below are all the steps you need to follow:

- ♦ Create a non-root user to run the node (more secure)
- ♦ Install needed softwares including Java
- ♦ Update the Linux system

1. First of all, you need to create a connection with the IP address and SSH port number of your VPS.



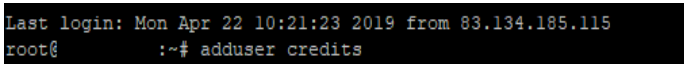
2. Log into your VPS for the first time as root.



Nothing is displayed on screen when you type your password, thus be attentive of what you're writing.

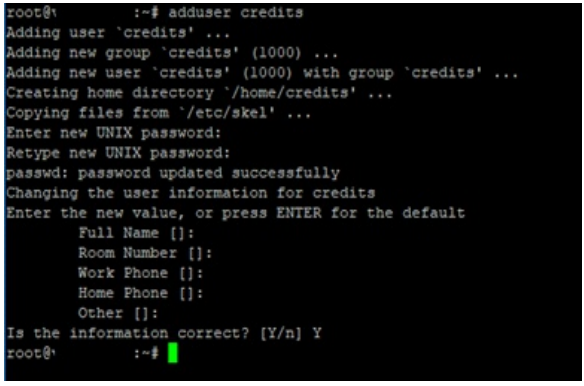
3. Add a user named "credits" (or any name you want). This user will run the node when installation will be done.

```
root@myvps:~# adduser credits
```



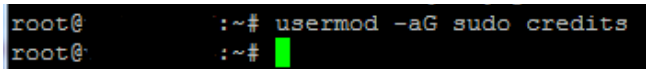
Only type commands that are shown in bold font. You can also make a copy-paste of those commands.

4. Write a password, confirm it and validate by entering other values.



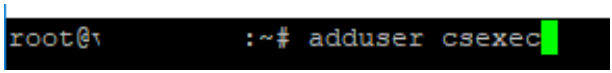
5. Now you are going to add the "credits" user in the sudo group, this will allow this user to install softwares needed and run the node. Type this command at the prompt:

```
root@myvps:~$ usermod -aG sudo credits
```

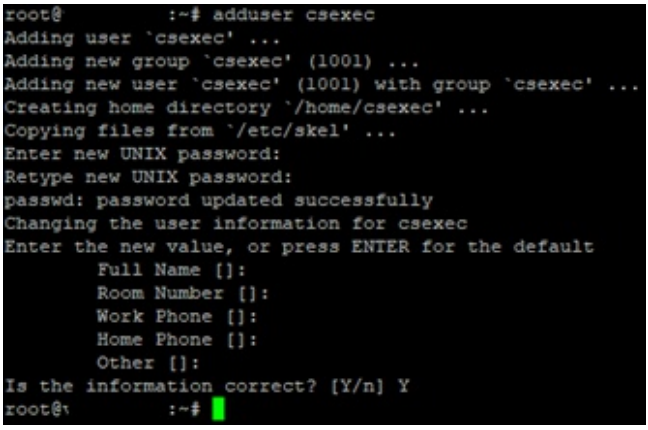


6. Add a user named "csexec" to run smart contracts in a secure/siled directory.

```
root@myvps:~# adduser csexec
```

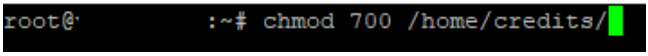


7. Then, write a password, confirm it and validate by entering other values.



8. Type the following command. Access to the /home/credits directory will be now limited to the main user "credit" only.

```
root@myvps:~# chmod 700 /home/credits/
```

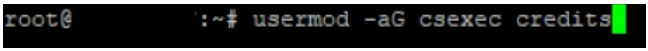


9. Then, you are going to give the user csexec and the csexec group permission to read and write in its home.

```
root@myvps:~# sudo chmod 770 /home/csexec
```

10. Add the credits user to the smart contract runtime group to put the corresponding .jar.

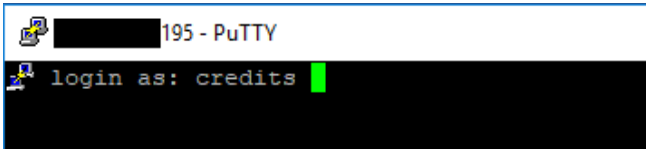
```
root@myvps: ~# usermod -aG csexec credits
```



11. Reboot to apply those changes.

```
root@myvps:~# shutdown -r now
```

12. Start PuTTY again, and reconnect to your VPS. It's more secure to connect only with the non-root user previously created (user "credits"), so log in with user "credits" you have just created, and associated password.



13. Now you're going to type the following commands to update the system and install a few tools + Java client. You can copy-paste the following commands into Putty and run them. Don't be surprised if the password of user 'credits' is asked once. Just type it.

```
credits@myvps:~$ sudo apt-get update

credits@myvps:~$ sudo apt-get upgrade
```

```
credits@myvps:~$ sudo apt-get install tmux

credits@myvps:~$ sudo apt-get install nmon

credits@myvps:~$ sudo apt-get install nload

credits@myvps:~$ sudo apt-get install htop

credits@myvps:~$ sudo apt-get install nano

credits@myvps:~$ sudo apt install net-tools

credits@myvps:~$ sudo apt install apport

credits@myvps:~$ sudo apt update

credits@myvps:~$ sudo apt install openjdk-11-jdk
```

STEP 5:

UPLOAD OF THE CREDITS SOFTWARE WITH FILEZILLA

In this part, you have to upload Credits node folder on your VPS. But first, you have to download it on your PC or Mac. Download the Linux node software on your computer from the developers portal.

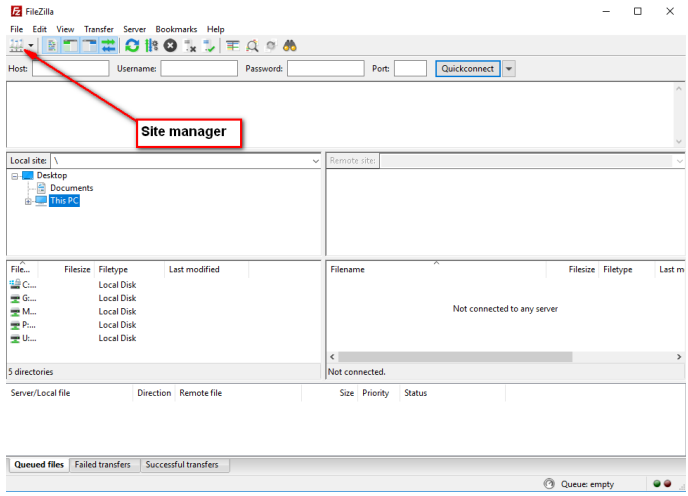
DOWNLOAD

With the FileZilla software installed on your computer, you are going to create a secure FTP connection (SFTP) on the Linux VPS in order to upload the **Credits_Node_linux_x64_ver_4_2.tar.gz** file (*file name will change, depending on the last version released by Credits*) to the home folder of the **"credits" user** previously created.

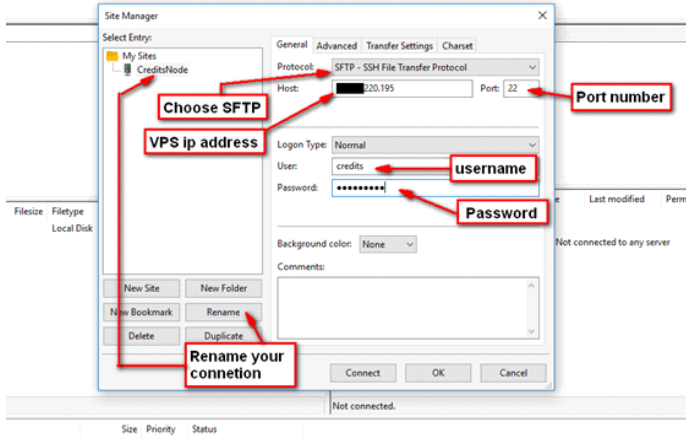
Copy the exact name of the Credits file you have just downloaded. You will have to modify the file name in some of our commands and replace it with your Credits software file name in order to avoid any issue.

Please follow steps in next page.

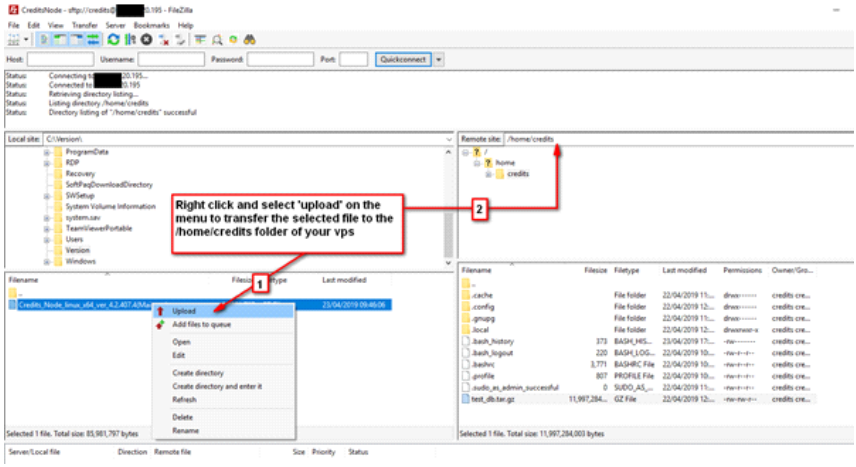
1. Run FileZilla and you will get opened the following window. Then click on the site manager (in "File" option).

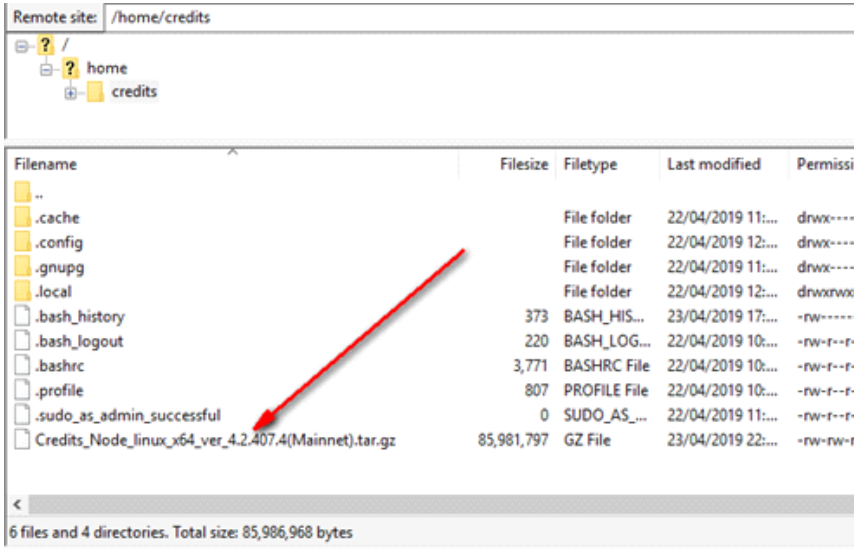


2. Follow the screenshot below to create a new connection and connect on your VPS.



3. Now that you are connected on your VPS through SFTP with FileZilla, you just have to upload the Credits folder on the */home/credits* folder.



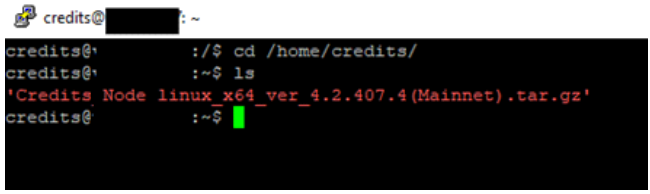


STEP 6:

EXTRACT THE CREDITS SOFTWARE FOLDER

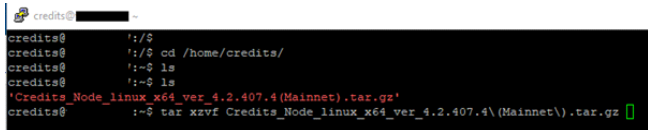
1. Once the CS node archive is uploaded on */home/credits/*, go back to PuTTY and move to the home folder of "credits" user in order to run those commands, which locates your uploaded Credits archive.

```
credits@myvps:~$ cd /home/credits/

credits@myvps:~$ ls
```



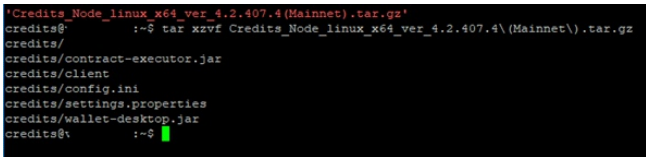
2. Now extract the Credits_Node_linux_x64_ver_4_2.tar.gz file

```
credits@myvps:~$
tar xzvf Credits_Node_linux_x64_ver_4_2.tar.gz
```



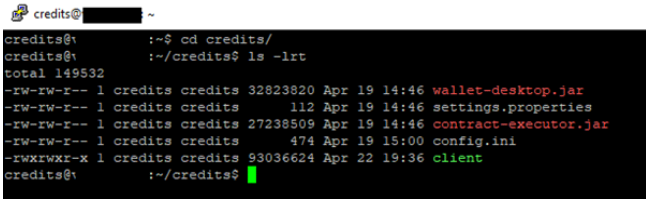
Remember that you will surely have to change the Credits software name while you type this command. It must match with your own current CS file name.

You will get this result:



3. Go into the extracted Credits folder and take a look at the content by typing:

```
credits@myvps:~$ cd credits

credits@myvps:~/credits$ ls -lrt
```



Informations in our tutorial screenshots may be slightly differents than informations shown in your screen, depending on the Credits node version you use.

If you want to copy-paste commands, just place your cursor at the right place in the line and make a right-click with your mouse, then press Enter.

Now that everything is in place to run the Credits node, you need first to be more familiar with Tmux because you are going to work now exclusively with it when connecting on your VPS with PuTTY SSH client. You will see in the next step that using Tmux is mandatory if you want to close your PuTTY session and let your Credits node run 24/7.

4. Now, move back to the parent directory by typing:

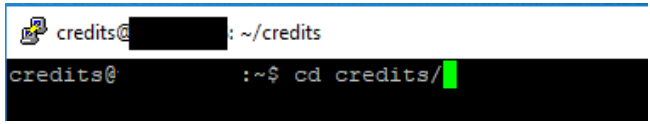
```
credits@myvps:~/credits$ cd ..
```

5. Copy the contract-executor.jar file and the settings.properties file to the home directory of csexec user.

```
credits@myvps:~$ sudo cp /home/credits/credits/contract-
executor.jar /home/csexec/
```

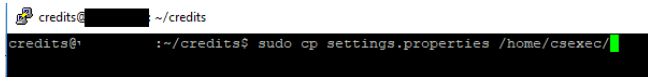
6. Move to the credits folder

```
credits@myvps:~$ cd credits/
```



7. Then copy the settings.properties file of credits user to the home folder of csexec user:

```
credits@myvps:~/credits$
sudo cp settings.properties /home/csexec/
```



Now that this settings are made, you are going to set your Credits node using Tmux (Step 7 and Step 8). Nevertheless, we strongly recommand you to take a look at our Step 10 (How to secure your VPS) BEFORE creating your node. Safety must be your priority!

STEP 7:
USING TMUX THROUGH PUTTY

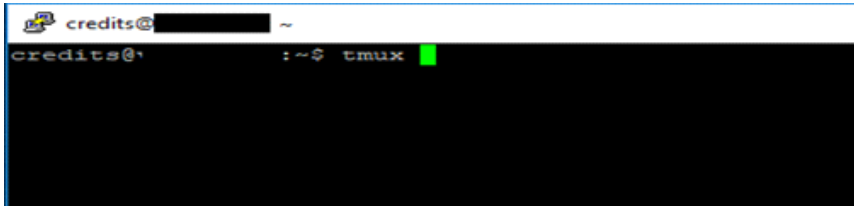
Now that everything is ready, you are going to launch **Tmux** through PuTTY. Tmux will allow you to run uninterruptedly your CS node and others programs/applications, and they will continue to run even if you close our PuTTY SSH client. Hence, next time you will connect on your VPS you will find your node and other apps running like the last time you have seen them running before closing PuTTY.

You are going to see in this part the basic functions needed to run the CS node, but below are some interesting tutorials about Tmux:

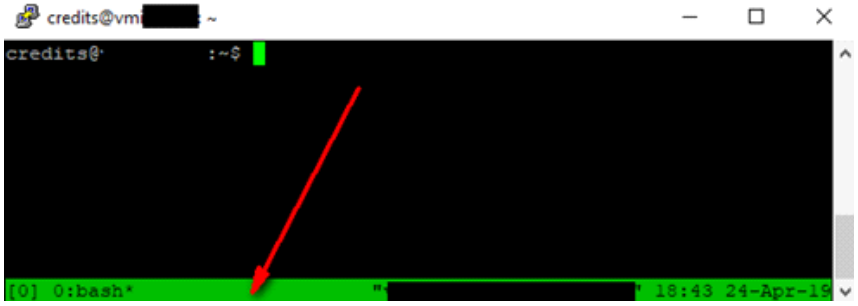
- **danielmiessler.com/study/tmux/**
- **hamvocke.com/blog/a-quick-and-easy-guide-to-tmux/**

Special characters and numbers from your numeric keypad are not taken into account by Tmux, thus you must use the main keyboard.

1. Run Tmux.



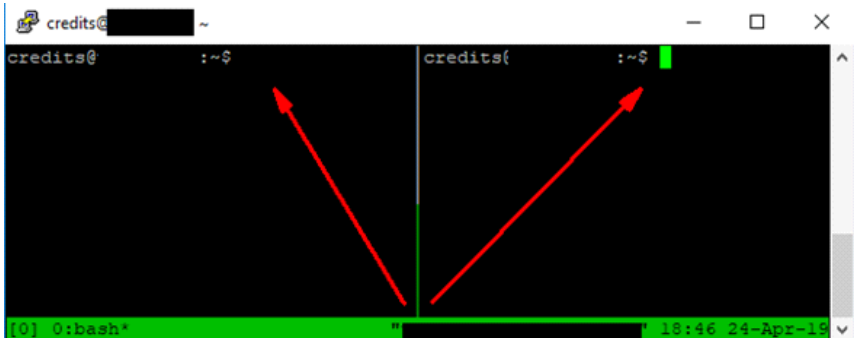
You can see that Tmux is running by viewing a green bar at the bottom of the SSH panel.



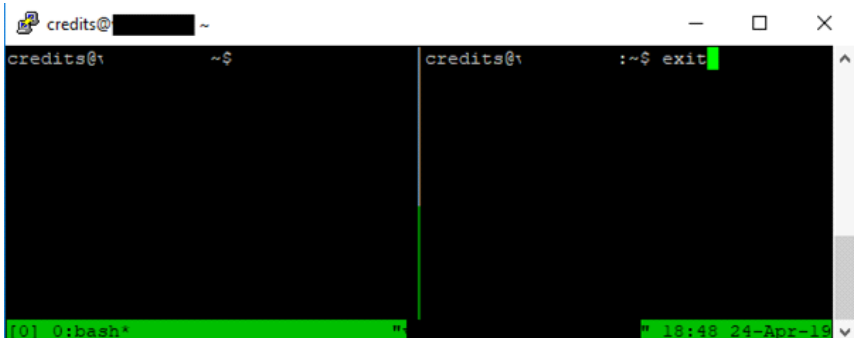
2. Now run: CTRL-B + % (don't type the [-]and the [+]).

Press CTRL key, don't release it and press B key. Then release the two keys and press %.

You will see that your panel has been split vertically.



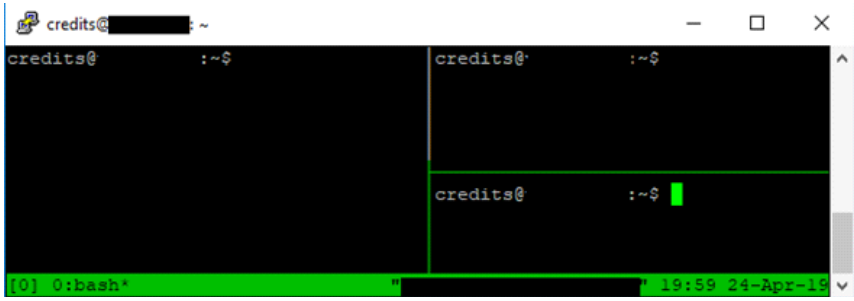
You can see that the active panel is on the right thanks to the green Tmux cursor. If you want to close this extra panel just created, you just have to type "**exit**".



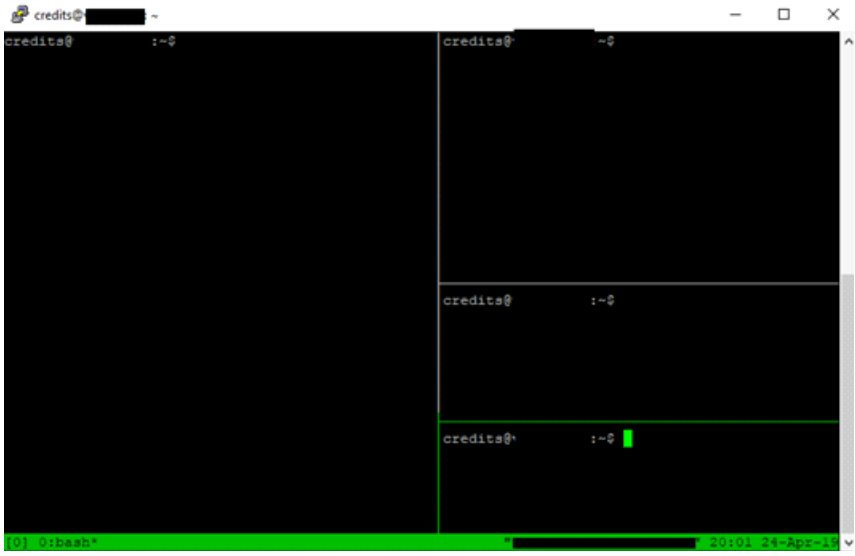
Result:

3. Now type again: CTRL-B + % (Don't type the [-]and the [+])

4. You can see that the curser is on the right panel. You are going now to divide the right-side panel horizontally by pressing: CTRL-B + ” (Don’t type the [-]and the [+])



5. Just enter one last time CTRL-B + ” and you will have enough panels to run node client, the contract executor and the monitoring tools.



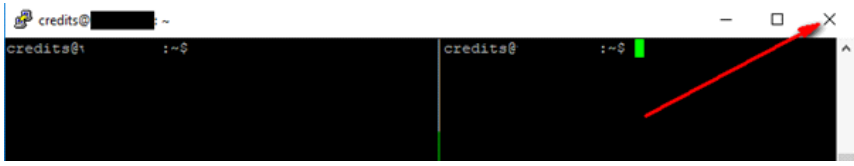
Navigating through panels

Switching to a different panel uses the **CTRL-B + <arrow key>** shortcut, where **<arrow key>** is the direction pointing to the panel you want to switch to. In your case, you want to switch to the left panel, so it's **CTRL-B + <left>**. Just once more, in order to fully understand this concept: this means you press CTRL and B (your prefix) followed by the left arrow key to get to the panel on the left.

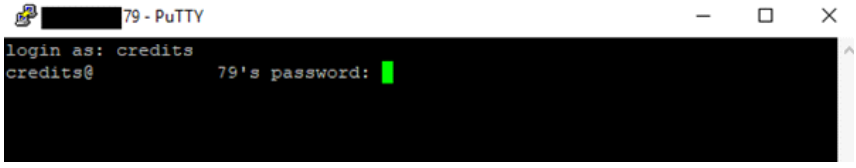
The main advantage of Tmux is that you can run your node and others tools, then quit your SSH session and when you log in again and run "tmux a" at the prompt, you will recover your Tmux session (and of course your CS node running!).

Let's try it!

1. Close your Putty SSH session:



2. Connect again on your VPS:

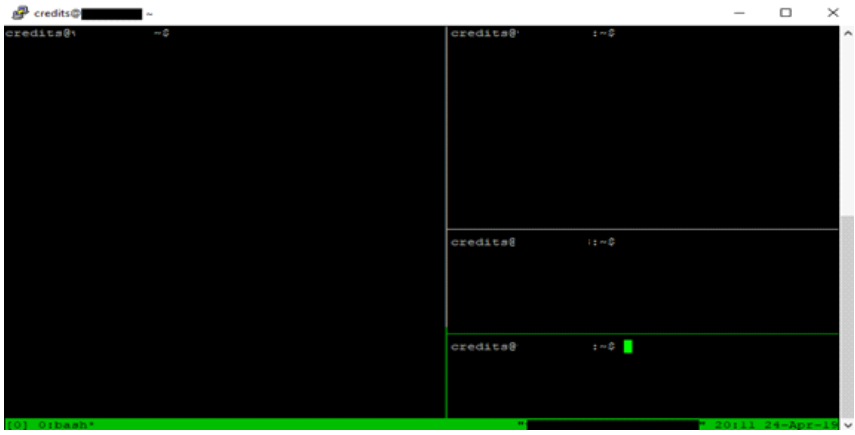


3. Once connected, just type:

```
credits@myvps:~$ tmux a
```

Your previous closed session will appear. In the screenshot below, uninterrupted Tmux session has been restored. You can see that there are only four empty panels, but if there was a CS Node or any other

programs running in one of those panels, they will be still running when typing "**tmux a**" at the prompt after connecting again with PuTTY.



STEP 8:
RUNNING THE CS NODE THROUGH TMUX

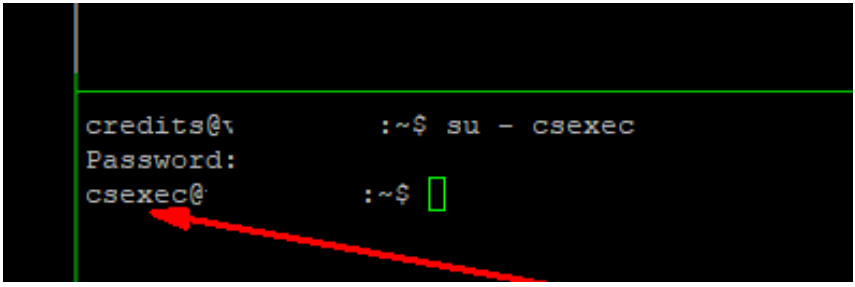
Let's begin with running the contract executor. In that case, you want to switch to the right-top panel, where you will create and open the contract executor. So, type twice **CTRL-B + <up>**.

1. STARTING THE CONTRACT-EXECUTOR.JAR

You have first to start the **contract-executor.jar** client before starting the node client. But for safety reasons, **the contract-executor.jar file must be ABSOLUTELY started with the csexec user specially created for this purpose.** In order to do it, type the following command at the prompt.

1. Type the following command and enter the password of user 'csexec':

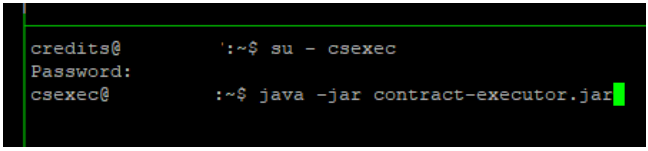
```
credits@myvps:~$ su - csexec
```



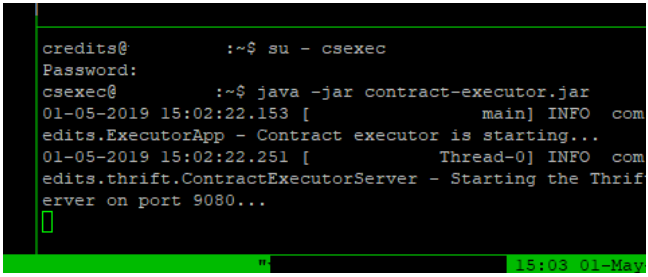
As you can see, we have switched user and are working as csexec user.

2. Then start the contract-executor with:

```
credits@myvps:~$ java -jar contract-executor.jar
```



The Contract-Executor will start:

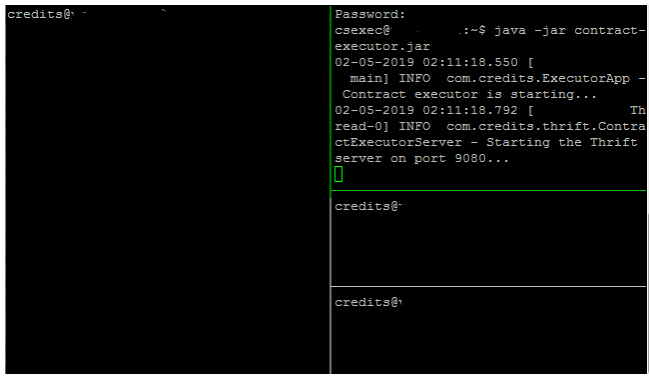


Every program running into the panels must be closed gracefully with CTRL+C. For example, if you reboot your server without closing the node client with CTRL+C, there is a great probability that you will crash your database and then restart synchronization from 0%. So please be cautious, hold CTRL and then press C!

If you just need to shut down your PC or Mac and want to let the node run, you don't have to close your panels with CTRL+C. Just close your windows with the cross.

2. STARTING CS CLIENT NODE

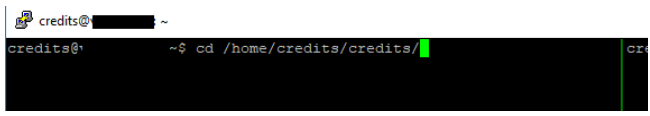
For now, your Tmux session looks like this:



You have contract executor running on the top right panel, and you are going to move on the left panel to start the CS node.

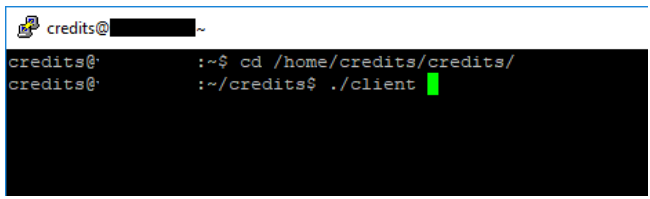
1. Hit CTRL-B <left> and when you're ready, type this:

```
credits@myvps:~$ cd /home/credits/credits/
```

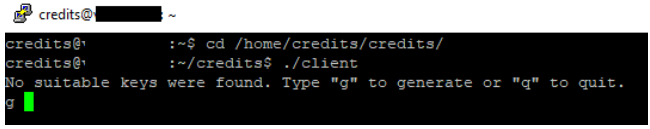


2. Then start the Credits client node with:

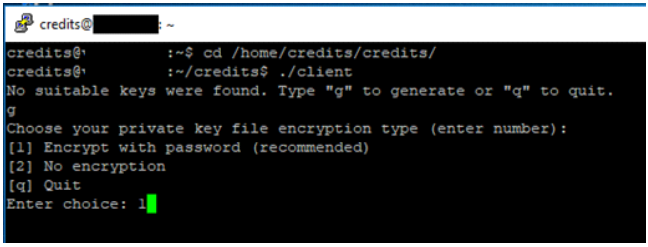
```
credits@myvps:~/credits$ ./client
```



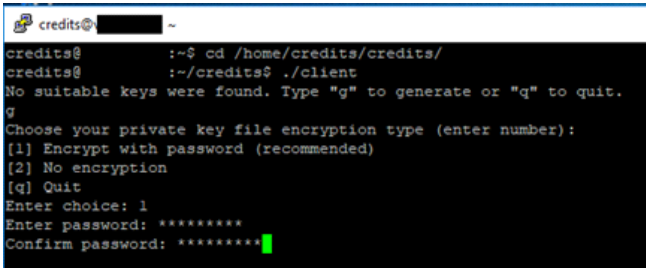
3. Type "g" to generate public and private keys, then press Enter.



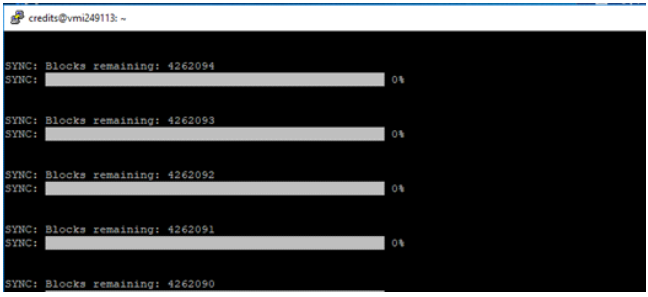
4. Then choose to Encrypt with password your private key file [1], followed by Enter.



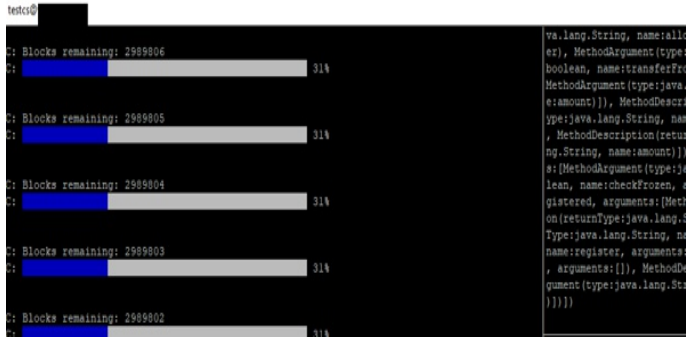
5. Enter your password, confirm it and type Enter.



Your CS node is starting, please be patient…



After a moment, you will see the sync process starting:



Congrats, you CS node is running!

If you want your node to be synchronized much faster, you can ask to Credits team for the latest database files, as well as *_integr.seq* file.
Once downloaded, you will have to upload this database files in */home/credits/credits* folder, then extract them, and finally replace current *_integr.seq* file by the one you will have downloaded separately.

3. PRIVATE AND PUBLIC NODE ADRESSES BACKUP

Now that your node is running, you have to backup your private and public node adresses.

Via FileZilla, go to your **Credits** folder, and you will see two files among the others: **NodePrivate.txt** and **NodePublic.txt.**

Copy them and paste them into a **secure folder outside your VPS (or best, on a sheet)**, you will need them to unlock your CS wallet.

4. OPENING NODE WALLET WITH ENCRYPTED PRIVATE KEY

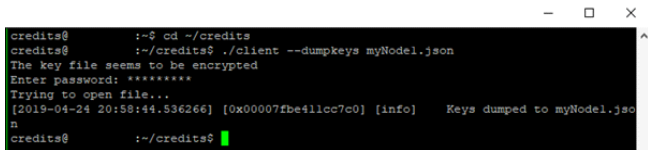
In **Part 2 ("Starting CS client node")**, we have choosen to encrypt private key. Therefore this key is protected and can only be used to run your node, **not to open your wallet.**

To open the node wallet, you need to generate the unencrypted key in a file. In this exemple, you will generate a file named **myNode1.json** but you can replace this name by the file name you want.

With **Tmux**, select a free panel (not running the contract executor or the client) or create a new one (**CTRL-B + %** or **"**), and type:

```
credits@myvps:~$ cd ~/credits

credits@myvps:~$ ./client --dumpkeys myNode1.json
```

Be careful, you must type a double dash just before *"dumpkeys myNode1.json"*



Enter your password and then hit **Enter**. This will generate the **myNode1.json** file in your **credits** folder. Now you can use **FileZilla** to copy this file on your own computer and save it in a safe place. This file can be directly used to open wallet client applications (Credits web wallet or desktop wallet). Open your wallet, click on **"Sign in",** then **"Upload file"**, and choose the **.json file** previously generated!

Last thing to do to be safe is to **delete this file on the VPS**. Go back to your Tmux terminal and type the following line:

```
credits@myvps:~$ rm myNode1.json
```

5. SENDING CS TO THE NODE WALLET

If you want to participate in transactions validation and **earn Credits rewards**, you have to send **more than 50.000 CS** to your node wallet (ie: 50.001 CS), and **stake it for as long as you want to earn CS rewards**.

Thus, you have to send this amount from your current wallet to the CS node adress written in your **NodePublic.txt** file. That's it!

NB: be sure that token swap occured before sending any CS coin.

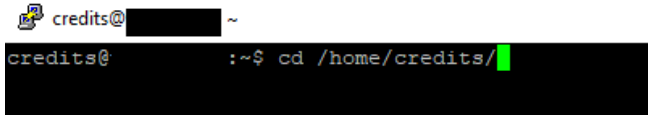
6. REINSTALLING A CREDITS NODE

If you want to reinstall your node, please first backup your files for a better safety (through FileZilla) and then follow this steps.

First you are going to delete your Credits folder.

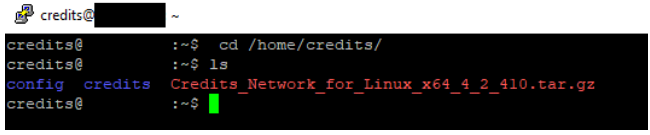
1. Move in /home/credits folder:

```
credits@myvps:~$ cd /home/credits/
```



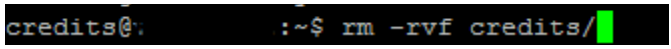
2. List the repertory:

```
credits@myvps:~$ ls
```



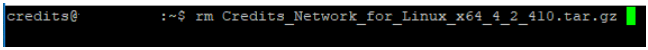
3. Delete Credits repertory and its content:

```
credits@myvps:~$ rm -rvf credits/
```



4. Delete your previous tar.gz archive (modify its name in the following command if needed)

```
credits@myvps:~$
rm Credits_Network_for_Linux_x64_4_2_410.tar.gz
```



5. Now that it's done, you just have to follow again the Step 5 (Upload of the Credits software with FileZilla) and Step 6 (Extract the Credits software folder) sections.

7. UPDATING A CREDITS NODE

If you want to update your Credits node with a new release without installing again the whole system, it's quite easy. Just follow this steps:

1. Download the new Credits node release from Credits website, as well as the new *__integr.seq* file (provided by Credits team).

2. With FileZilla, upload those two files into your */home/credits* folder.

3. Go back to PuTTY and shut down your current client and contract executor with CTRL+C (you can then type *exit*).

4. Connected as *credits* user, type into an empty Tmux panel:

```
credits@myvps:~$ cd ~/
```

5. If, and only if, a previous backup folder named *credits_old* has already been created (you can check it with *ls -lrt* command, or in FileZilla), delete it by typing:

```
credits@myvps:~$ rm -rf credits_old/
```

6. Create a backup of the actual *credits* folder by renaming it into a folder named *credits_old*:

```
credits@myvps:~$ mv credits/ credits_old

credits@myvps:~$
tar xzvf Credits_Node_linux_x64_ver_4.2.411.tar.gz
```

7. Then, we move the NodePublic.txt and NodePrivate.txt files located in the *credits_old* folder into the freshly *credit* extracted folder:

```
credits@myvps:~$ mv ./credits_old/NodeP* ./credits/
```

8. We also get back the database:

```
credits@myvps:~$ mv ./credits_old/test_db/ ./credits/
```

9. Then we update the versionning database file (*__integr.seq*):

```
credits@myvps:~$ mv ./__integr.seq ./credits/test_db/
```

10. We update contract executor for *csexec* user:

```
credits@myvps:~$ cd credits

credits@myvps:~$
sudo cp ./contract-executor.jar /home/csexec/
```

11. Now, you can relaunch the contract executor as a *csexec* user (see Step 8.1), and the updated Credits client with as a *credits* user (see Step 8.2)!

STEP 9:
RUNNING THE MONITORING TOOLS

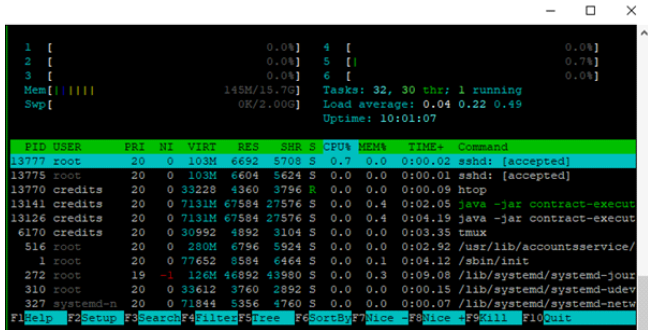
Under Linux, you have a countless list of more powerful monitoring tools than any other. You are of course free to choose to use all those who will suit you. In this tutorial, we will introduce four tools:

- **htop:** overview and manage running process on the VPS
- **nmon:** indicate the processor load
- **nload:** check the bandwidth used
- **df command:** indicate the free space on your disk

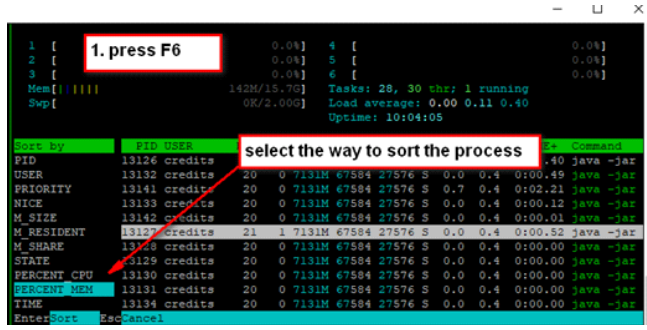
HTOP

Move to an unused panel and type:

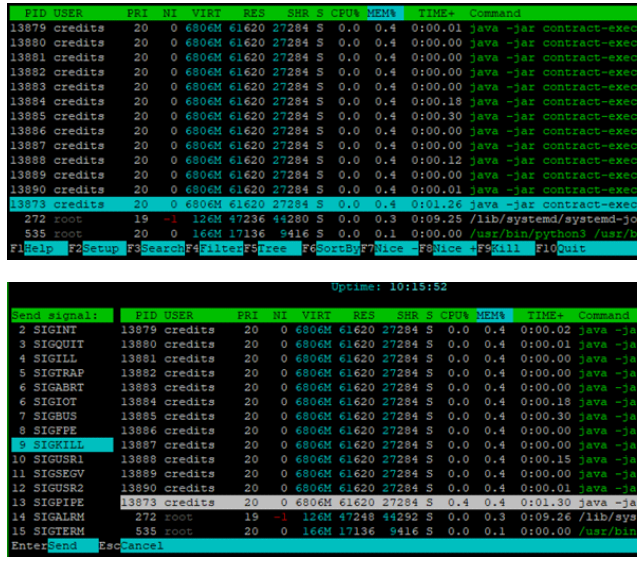
```
credits@myvps:~$ htop
```



You will be able to sort all process by pressing a function key. For example, if you press **F6** function key, a panel will appear on the left side of the htop panel, and it will allow you to sort the process by memory consumption.



You can also directly see the uptime, cpu load, free memory. **But one of the most useful function with htop is to be able to stop/kill a process if needed.** You just have to highlight in blue a process (in this example, the java contract executor with PID 13873) and then press the **F9** function key. A menu will also appear on the left side and just move to option **'9 SIGKILL'** and press **Enter** to kill the selected process.

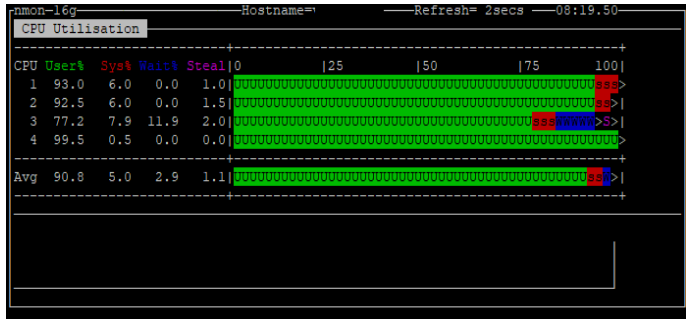


NMON (CPU LOAD)

Move to an unused panel and type:

```
nmon
```

Then type "**C**" key, and you will be able to see your VPS CPU load.

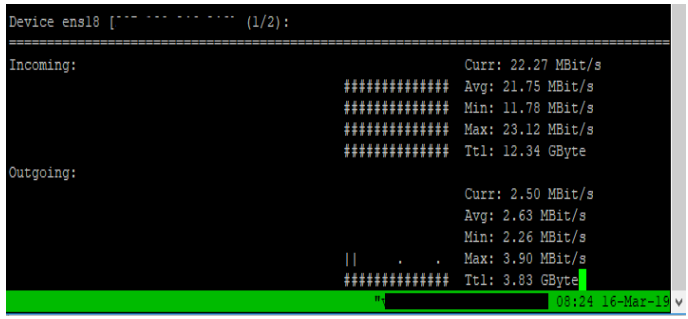


NLOAD (NETWORK TRAFFIC)

Move to another unused panel and type:

```
nload
```

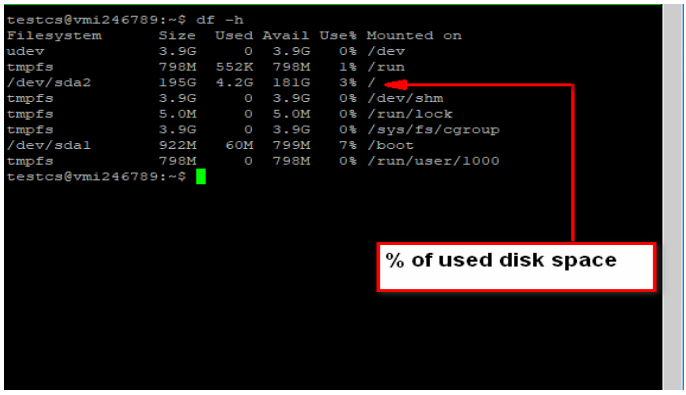
You will then see the network traffic of your VPS:



DF COMMAND (DISK SPACE)

You can also use the "**df**" command to visualize the disk space used by the CS database, which will grow with time.

```
df -h
```



USEFUL TMUX COMMANDS

1. If you want to list your current sessions:

```
tmux ls
```

2. If you want to access to the right session (replace "myname" by your session name):

```
tmux a -t myname
```

3. If you want to visit your different sessions:

```
tmux w
```

4. If you want to restart programs from the beginning:

```
sudo reboot
```

5. If you want to kill a specific session (replace "myname" by your session name):

```
tmux kill-session -t myname
```

6. Then, the following line will be enough to access to your current session:

```
tmux a
```

7. If you want to open another Tmux window

```
CTRL-B + C
```

8. If you want to close a Tmux window

```
CTRL-B + &
```

USEFUL TMUX SCRIPT

Tmux CS session

If you want to detect if there is already a Tmux session running, or if not, if you want that it creates automatically a new one with CS node, contract executor and nmon, you can use this bash script (assuming that your install is in **/home/credits/credits** and **"nmon"** is installed).

This script only works if you have chosen to run contract executor as *credits* user, and not as *csexec*.

1. Download the bash script (Dropbox link):

DOWNLOAD

2. If your installation is not in "/home/credits/credits", open the file with a free editor like Notepad++ (or vi, for more advanced users), and modify lines 12 and 15 with your own path.

```
1  #!/bin/bash
2
3  function has-session {
4    tmux has-session -t credits 2>/dev/null
5  }
6
7  if has-session ; then
8        echo "Session already exists"
9     tmux -2 new-session -A -s credits -c credits
10 else
11    echo "Session credits does not exist... create it"
12    tmux -2 new-session -d -s credits -c /home/credits/credits ./client
13    tmux rename-window 'CS node'
14    tmux set-window-option -t credits remain-on-exit on
15    tmux split-window -h -p 40 -c /home/credits/credits java -Xmx3G -jar contract-executor.jar
16    tmux split-window -v -p 50 'nmon'
17    tmux send-keys 'cu'
18    #tmux selectp -t:0.1 -P 'bg=red'
19    tmux new-window -n 'bash' bash
20    tmux select-window -p
21    tmux attach-session -d -t credits
22 fi
23 exit 0
```

Be careful when you save your file! You need to "Save as…" and choose "Unix script file" format.

3. Now open PuTTY, and type the following commands :

```
credits@myvps:~$
chown credits. /home/credits/tmux-cs-session.sh

credits@myvps:~$
chmod u+x /home/credits/tmux-cs-session.sh

credits@myvps:~$
cd /home/credits/

credits@myvps:~$
./tmux-cs-session.sh
```

4. The next time you will want to access to your Tmux session, you will have to type the following command at the beginning of PuTTY (of course once you are logged in).

```
credits@myvps:~$ ./tmux-cs-session.sh
```

If your session is already created, the script doesn't recreate another one and will connect on the existing session.

If your session is not created yet, it will create a new one with CS node, contract executor and nmon.

That's it!

Creating an apport application to get the core dumped file

The apport application automatically reports only managed packaged applications. When application crash, the core dump files are put in **/var/crash**.

The Credits client app is not a packaged application, so let's configure apport to get the core dump file. It helps Credits team to identify issues when you get some troubles with your node.

1. Create and edit an apport settings file in .config

```
credits@myvps:~$ sudo mkdir -p ~/config/apport

credits@myvps:~$ sudo nano ~/config/apport/settings
```

2. Put this in the settings file

```
[main]

unpackaged=true
```

Save changes by **CTRL +X**, type **Y** and then press **ENTER**.

3. Restart apport

```
$ sudo service apport restart
```

4. After a crash, unpack the crash file generated by apport:

```
credits@myvps:~$ cd /var/crash

credits@myvps:/var/crash$ mkdir /tmp/report

credits@myvps:/var/crash$ apport unpack _home_credits_cre
dits_client.1000.crash /tmp/report
```

The previous line has to be written in a single line, beginning with "apport-unpack", then SPACE, then "_home_credits_credits_client.1000.crash /tmp/report"

5. Then, type this last line:

```
credits@myvps:/var/crash$ ls -l /tmp/report
```

Then you will have your core dumped file (it can take a while to unpack: ~1 or 2 min).

NB: The pattern of the core dumped file is here. You can modify it if you want, but it's not recommended:

```
credits@myvps:~$ cat /proc/sys/kernel/core_pattern

|/usr/share/apport/apport %p %s %c %d %P
```

STEP 10 :
HOW TO SECURE YOUR VPS

It's very important to secure your VPS against malicious attacks. To do this, we will implement in this chapter tools that will increase the level of security.

First of all, it is very important to run these two commands as regularly as possible to keep your system up to date.

```
credits@myserver:~$ sudo apt-get update

credits@myserver:~$ sudo apt-get upgrade
```

Secundly, it is very important that your Credits node is shutted down while you follow this safety steps. It's also recommanded to do this actions before setting the node or at least, to have a backup of json file and Nodepublic and Nodeprivate files.

1. CHANGE THE SSH DEFAULT PORT OF YOUR VPS

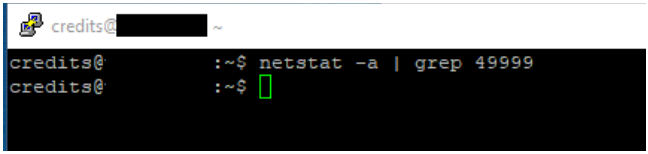
This part will explain how to change the default SSH port in your Linux VPS. By default, the SSH is listening on port 22. This change of the SSH port will add an extra layer of security and will stop many attacks.

You have to choose a port between **49151** and **65535** (those are dynamic/private ports and can be used).

Let's assume that we are going to use the port **49999**. To check if that port is used, type the following command:

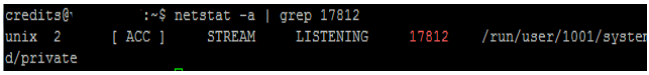
```
credits@myvps:~$ netstat -a | grep 49999
```

If the port is not used, you should not get any results:



That means that the port is free. But if you type the same command with the port **17812** for example, you will get this result:

```
credits@myvps:~$ netstat -a | grep 17812
```

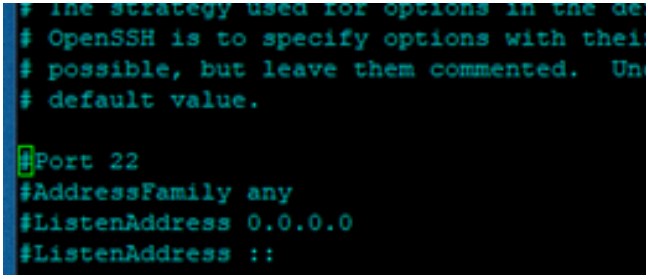


You can see that the port 17812 is LISTENING.

Now, we can edit the SSHD configuration port in order to assign the new port selected.

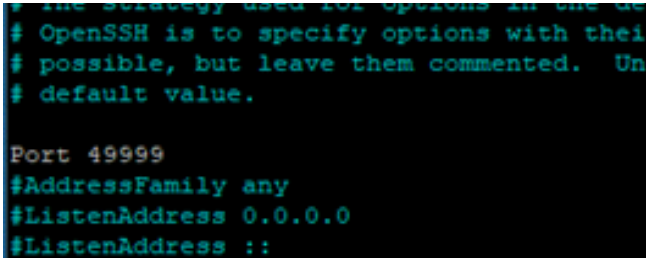
```
credits@myvps:~$ sudo nano /etc/ssh/sshd_config
```

Uncomment this line (by removing **#**) and change 22 by **49999**.



Please note that by this default SSH port change, the new selected port will also be the one you will need to use during a SFTP connection (FileZilla).

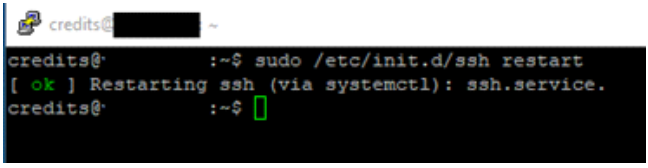
It should looks like this:



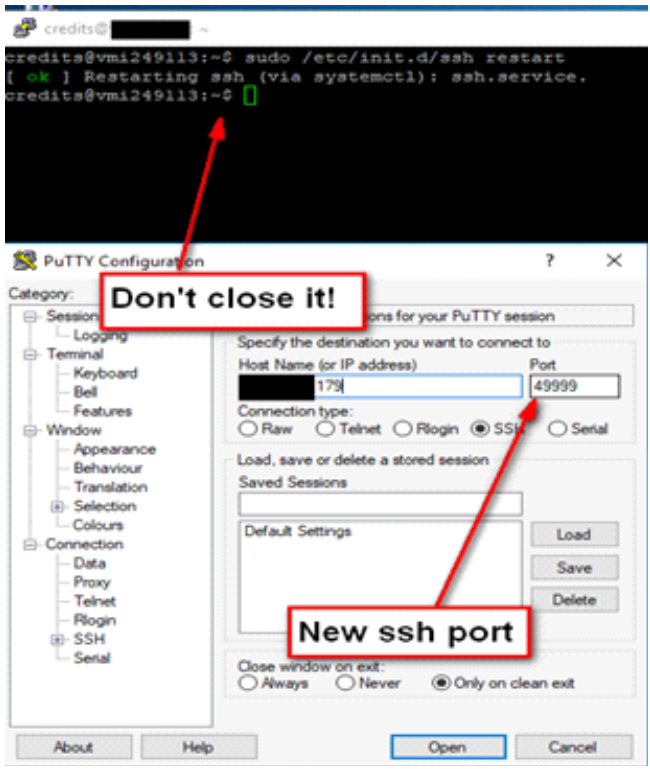
Save changes by **CTRL+X**, type **Y** and then **ENTER**. By security, open again your SSHD_Config file to see if the changes are saved (very important, otherwise you take the risk to loose the connectivity to your VPS for good).

Restart the SSH service to switch over to the new port 49999:

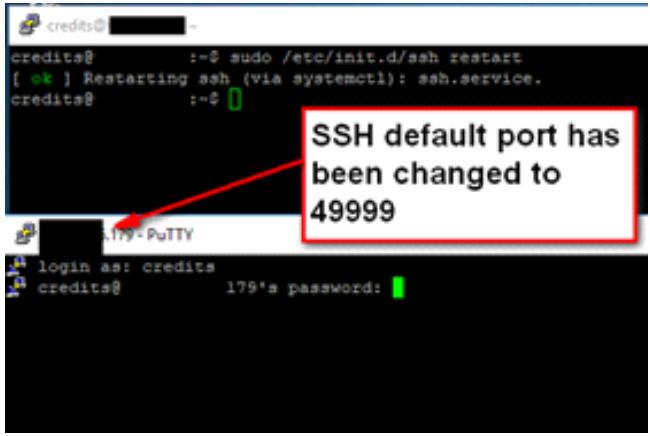
```
credits@myvps:~$ sudo /etc/init.d/ssh restart
```



Before closing the actual SSH session, open a new one and test the new port. It will allow you to make an undo of your changes in case of need! (see next picture)



That's ok. The new ssh port is now set to **49999**.



You can see the change by using the **netstat commands**. Port 22 is unused and 49999 is listening.

```
credits@myvps:~$ netstat -a | grep 49999

credits@myvps:~$ netstat -a | grep 22
```

2. FAIL2BAN

First of all, using a strong password to connect to your VPS is very important. Typically, **Fail2Ban** looks for repeated attempts to log unsuccessful log files, and proceeds to a ban by adding a rule to the IP tables firewall to ban the IP address of the source.

Fail2Ban is not really a security tool. The main goal is to avoid overloading the system logs with thousands of login attempts. A server with SSH access on the standard port, for example, will receive very quickly hundreds or even thousands of connection attempts from different machines. These are usually brute force attacks launched by bots.

Fail2Ban, by analysing the logs, makes it possible to ban the IP after a certain number of attempts which will limit the filling of the logs and the use of the bandwidth. But this does not improve the security of the service concerned. If the SSH access is not secure enough (weak password for example), Fail2Ban will not prevent an attacker from getting his way.

1. Installing Fail2Ban:

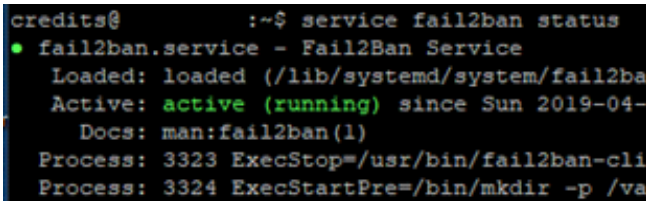
```
credits@myvps:~$ sudo apt-get update

credits@myvps:~$ sudo apt-get install fail2ban -y
```

2. After installation, start the service:

```
credits@myvps:~$ sudo service fail2ban restart
```

3. Make sure that the Fail2Ban service is running by typing:

```
credits@myvps:~$ service fail2ban status
```



4. Now that Fail2Ban is active and running, we need to create a configuration file for Fail2Ban, in order to define some custom rules and how to handle violations. Create the Jail.local file configuration file:

```
credits@myvps:~$ sudo nano /etc/fail2ban/jail.local
```

This file is empty, paste the following lines and save changes by **CTRL+X**, type **Y** and then **ENTER**. To be sure, open again your Jail.local file to see if the changes are saved by running the same command and by exiting with **CTRL +X**.

```
[DEFAULT]
ignoreip = 127.0.0.1/8 ::1
findtime = 600
bantime = 3600
maxretry = 5
```

Ignoreip: Fail2Ban will ignore the IPV4 and IPV6 of the localhost.

Findtime: time interval within the retry are counted.

Bantime: these rules will ban IP address for one hour.

Maxretry: maximum number of wrong attempts

That means that an IP address who try to connect with a wrong password a maximum of 5 times **(maxretry=5),** within 10 minutes (**findtime=600 sec**) will be banned for one hour (**bantime=3600**).

3. ENABLE TWO-FACTOR AUTHENTICATION ON A VPS (2FA)

Two-Factor Authentication (2FA) is an additional security measure that you can use on your VPS. Beyond from entering username and password, connecting to your server via SSH will require to enter a token from the Google Authenticator app.

1. Installation of Google Authenticator (GA):

You will need to install the Google Authenticator app on your mobile phone at first. Once done, follow these instructions:

Connect to your VPS using PuTTY (not with Tmux, as there are issues with GA options display). Then run the following commands:

```
credits@myvps:~$ sudo apt-get update

credits@myvps:~$
sudo apt-get install libpam-google-authenticator
```

Now, we need to generate a Time-Base On-Time Password. Type the following command and then put your window in full screen:

```
credits@myvps:~$ google-authenticator
```

Take notice of your secret key and of your emergency scratch code.

Scan the QR code with your smartphone within the Google authenticator app.

Answer YES to the following questions:

- Do you want me to update your "/home/credits/.google_authenticator" file? (y/n)

- Do you want to disallow multiple uses of the same authentication token? This restricts you to one login about every 30s, but it increases your chances to notice or even prevent man-in-the-middle attacks (y/n)

- In order to compensate for possible time-skew between the client and the server, we allow an extra token before and after the current time. This allows for a time skew of up to 30 seconds between authentication server and client. If you experience problems with poor time synchronization, you can increase the window from its default size of 3 permitted codes (one previous code, the current code, the next code) to 17 permitted codes (the 8 previous codes, the current code, and the 8 next codes). This will permit for a time skew of up to 4 minutes between client and server. Do you want to do so? (y/n)

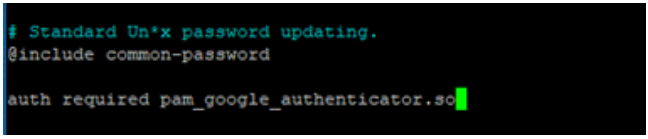
- If the computer that you are logging into isn't hardened against brute-force login attempts, you can enable rate-limiting for the authentication module. By default, this limits attackers to no more than 3 login attempts every 30s. Do you want to enable rate-limiting? (y/n)

2. Enable PAM on SSH:

We need to configure the **/etc/pam.d/sshd** file in order to allow user to connect using 2FA through SSH.

```
credits@myvps:~$ sudo nano /etc/pam.d/sshd
```

Add the line 'auth required pam_google_authenticator.so' at the end of the file:

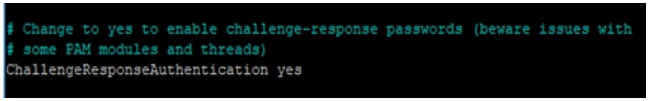


Then press **CTRL+X**, **Y** and **Enter**.

Now open the SSH config file to enable this way of authentication:

```
credits@myvps:~$ sudo nano /etc/ssh/sshd_config
```

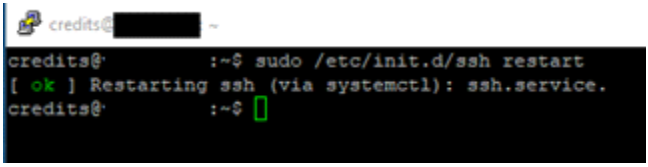
Then change the value of the line:
'**ChallengeResponseAuthentication**' from **no** to **yes**:



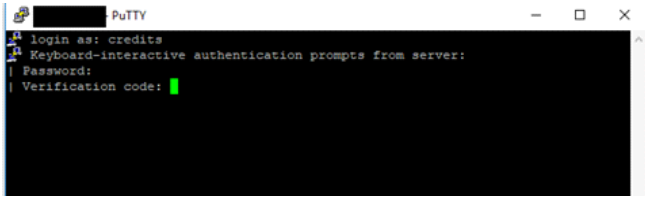
Save changes by **CTRL+X**, type **Y** and then **ENTER**.

Restart the SSH service:

```
credits@myvps:~$ sudo /etc/init.d/ssh restart
```



Quit your SSH session and connect again. You will see that the password is asked at first and then a 2FA code is generated by your Google Authenticator on your smartphone.



And one last cool thing is that FileZilla will also ask you a 2FA code after entering your password to connect through SFTP!

HOW TO RUN A NODE ON YOUR PC

STEP 1:

DOWNLOAD AND INSTALLATION OF THE CREDITS SOFTWARE

In this part, you have to download Credits software, extract it and install it on your Personal Computer. Mac users can't currently run a node on their computer.

1. Download and install JRE (Java Runtime Environment)

DOWNLOAD

2. Download the Credits software:

DOWNLOAD

3. Extract it with your extractor tool

4. Click on "exe" file and run it

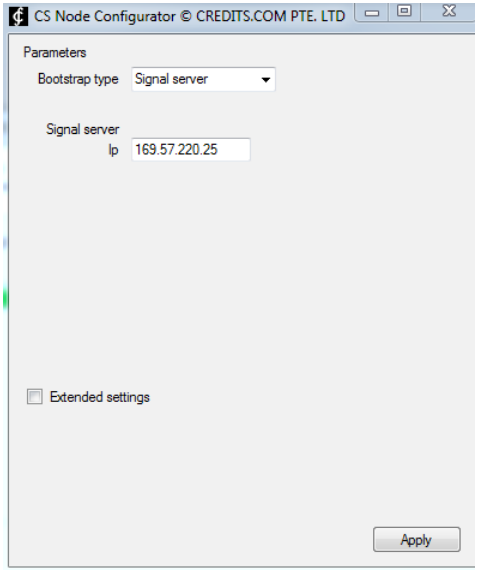
5. After having chosen your installation location, click on "Next" and then "Install"

STEP 2:
SETUP YOUR CREDITS NODE

Now that your Credits software is installed, you will have few things to set before the Credits node will be running on your computer.

1. SETTING CREDITS SOFTWARE

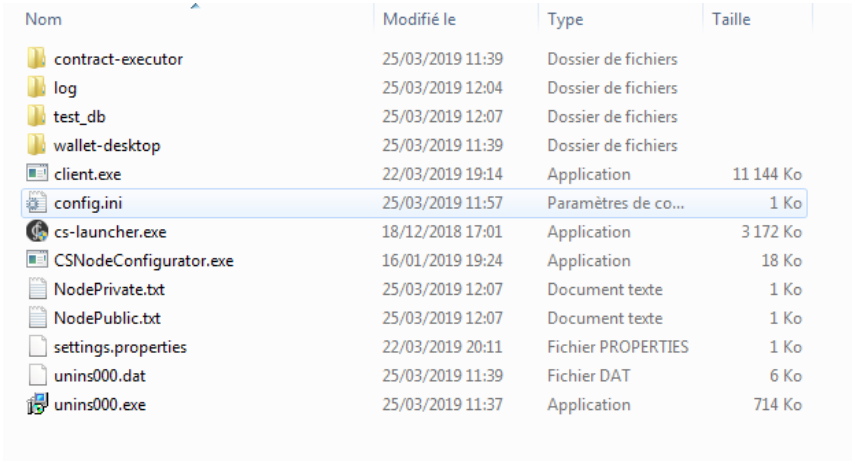
1. After having installed the software, a new window will appear. Let unchanged the default settings and click on "Apply".



2. Close the window and click on "Finish".

If you get any issue after running your node, you can try the following step.

Before launching Credits software, go to your new Credits folder (which is probably stored in your "**C**" disk), and open "**config.ini**" file with a text editor like **Notepad++**.



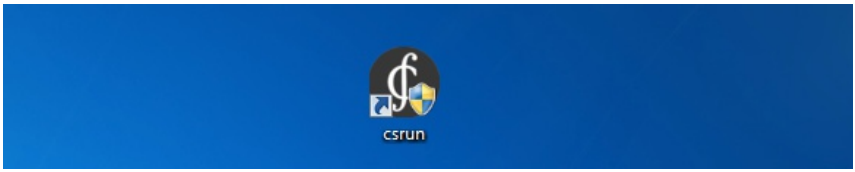
Nom	Modifié le	Type	Taille
contract-executor	25/03/2019 11:39	Dossier de fichiers	
log	25/03/2019 12:04	Dossier de fichiers	
test_db	25/03/2019 12:07	Dossier de fichiers	
wallet-desktop	25/03/2019 11:39	Dossier de fichiers	
client.exe	22/03/2019 19:14	Application	11 144 Ko
config.ini	25/03/2019 11:57	Paramètres de co...	1 Ko
cs-launcher.exe	18/12/2018 17:01	Application	3 172 Ko
CSNodeConfigurator.exe	16/01/2019 19:24	Application	18 Ko
NodePrivate.txt	25/03/2019 12:07	Document texte	1 Ko
NodePublic.txt	25/03/2019 12:07	Document texte	1 Ko
settings.properties	22/03/2019 20:11	Fichier PROPERTIES	1 Ko
unins000.dat	25/03/2019 11:39	Fichier DAT	6 Ko
unins000.exe	25/03/2019 11:37	Application	714 Ko

Delete the following lines, save the file, and restart your node:

```
[host address]
port = 6000
```

```
[params]
node_type=client
bootstrap_type=signal_server
ipv6=false

[signal_server]
ip =169.57.220.25
port = 6000

[host_input]
port=6000

[host_address]
port=6000

[api]
;port=9090
;executor_port=9080
apiexec_port=9070
;ajax_port=8081

[Core]
Filter="%Severity% >= info"
```

3. Now, it's time to launch Credits software. Right-click on "cs-launcher.exe" shortcut and run it as administrator.



Several windows will open, Credits node, Credits wallet and Java programs including Contract executor:

4. Show CS window. You will be asked to generate public and private keys. Press "G" key to generate them.

5. Encrypt your keys with a password, and confirm it. Then, press "Enter". Your node will begin its synchronization with Credits blockchain!

2. PRIVATE AND PUBLIC NODE ADRESSES BACKUP

Now that your node is running, you have to backup your private and public node adresses.

Go back to your **Credits** folder, and you will see two files among the others: **NodePrivate.txt** and **NodePublic.txt**.

Copy them and paste them into a **secure folder outside your PC (or best, on a sheet)**, you will need them to unlock your CS wallet.

3. OPENING NODE WALLET WITH ENCRYPTED PRIVATE KEY

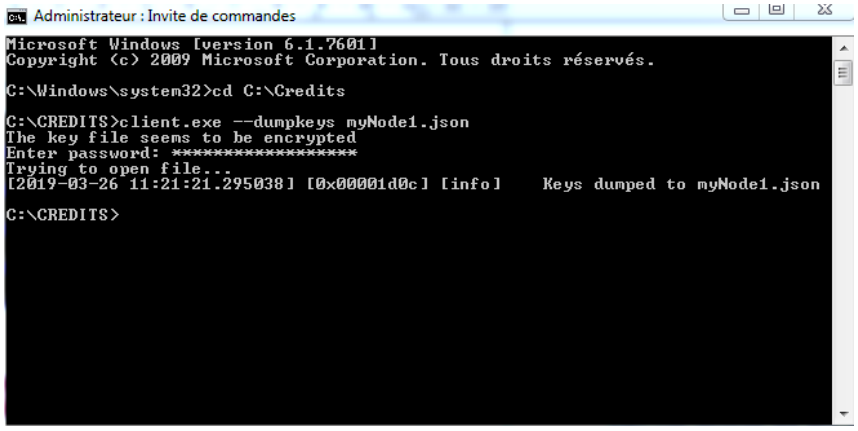
In **Part 1 ("Setting Credits software")**, we have choosen to encrypt private key. Therefore this key is protected and can only be used to run your node, **not to open your wallet.**

To open the node wallet, you need to generate the unencrypted key in a file. In this exemple, you will generate a file named **myNode1.json** but you can replace this name by the file name you want.

Run Command prompt as Administrator by right-clicking on it and selecting the **"Run as administrator"** option. Then type:

```
cd C:\Credits

client.exe --dumpkeys myNode1.json
```

Be careful, you must type a double dash just before *"dumpkeys myNode1.json"*. Please also be sure that Credits folder is located in "C:\" disk. If not, modify location in the first code line.



Enter your password and then hit **Enter**. This will generate the **myNode1.json** file in your **credits** folder. Now you can go to your

Credits folder to copy this file and save it in a safe place. This file can be directly used to open wallet client applications (Credits web wallet or desktop wallet). Open your wallet, click on **"Sign in",** then **"Upload file"**, and choose the **.json file** previously generated!

Last thing to do to be safe is to **delete this file from the Credits folder**.

4. SENDING CS TO THE NODE WALLET

If you want to participate in transactions validation and earn Credits rewards, you have to send **more than 50.000 CS** to your node wallet (ie: 50.001 CS) and stake it for as long as you want to earn CS rewards.

Thus, you have to send this amount from your current wallet to the CS node adress written in your **NodePublic.txt** file. That's it!

5. CLOSING CREDITS NODE

If you need to close your Credits node, please open the node window and use the **"CTRL+C" shortcut** in order to close it properly. It will avoid database corruption when you will open it again.

6. UNINSTALLING AND REINSTALLING CREDITS SOFTWARE

If you want to uninstall your Credits software and reinstall it with for example the Mainnet version, you have to go to your Control panel, select **"Uninstall program"** and choose **Credits** one.

Once uninstallation is finished, go to your Credits folder and just delete it.

You can now reinstall your new Credits software version.