Jupiter node manual

Client installation on Windows, MacOS, Linux VPS & Raspberry Pi and sources for general wallet information

V2.3 (September 2021)

Created by Letstrythis
Reviewed by Sigwo

Table of Contents

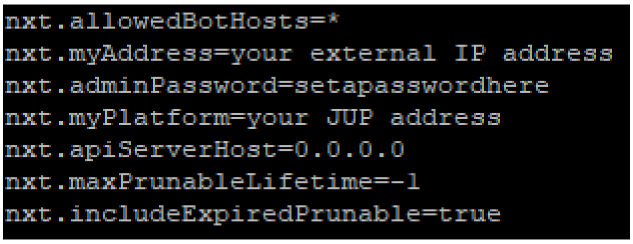
1 Client installation on Windows

1. To run the JRS client, Java 8 must be installed.
   You can download it from https://www.java.com.
2. Go to https://github.com/jupiter-project/jupiter/releases and download the latest JRS version.
3. Extract the downloaded zip file by right-clicking or using your preferred program.
   You can leave the extracted JRS folder in your download folder or can move it to your Program Files folder.
4. To start the client, go into the JRS folder and double-click on the run.bat file.
   If the client does not launch automatically, open your web browser and go to http://localhost:7876 or https://127.0.0.1:7876 to start it.
5. In (the hidden folder) C:\Users\$USER\AppData\Roaming\JRS a folder called nxt_db has been created. This is the folder where the blockchain database is stored. If you run the client, the blockchain will be downloaded automatically. This may take some time and to speed up this process you can choose to use the snapshot available on the Jupiter toolkit website:
   - Stop the client by closing both the wallet and Java windows.
   - Download the mainnet_longnumber.zip file from http://node-eu-de.jupitertoolkit.com/latest_db.php?download.
   - Extract the zip file and move the nxt_db folder to C:\Users\$USER\AppData\Roaming\JRS (replace your existing nxt_db folder with the extracted one).
   - Now you can restart the wallet.
     Note: when you have not used the wallet for a while and open it, in the bottom right a downloading message appears. This message does not update in real-time. To get rid of the message, log off and log into the wallet again.

   Now you are a proud Jupiter node owner.
   At the moment you are a “passive” node. This means you connect to the network and can use the wallet, but you are only downloading information. If you are a bit more experienced and especially if you are forging and have your node running 24/7 it would be nice to consider becoming an active peer. This means, besides outgoing connections to download blockchain data from other nodes, you will allow incoming connections so that other nodes can download blockchain data from you. This will increase the strength and security of the network. The only requirement is that you have a static, individual IPv4 address (so if your ISP for example makes use of CGNAT you can’t). For incoming peer-to-peer networking requests, JRS uses port 7864. This means your firewall and router should allow this traffic.
6. When using the JRS software for the first time, Windows Firewall will ask you to allow the software having incoming and outgoing connections. After accepting, Java(TM) Platform SE binary is allowed to have incoming traffic from any port, so your firewall does not need any further configuration.
7. You have to open port 7864 for incoming TCP traffic on your router and configure port forwarding to your computer that runs JRS. These steps are router specific so just Google “your specific router/modem + port forwarding” if you need any help. Some general info can also be found on websites like https://portforward.com and https://www.noip.com/support/knowledgebase/general-port-forwarding-guide. You can check whether you’ve configured everything successfully by looking up port 7864 for your IP address on https://portchecker.co. It should say that the port is open.
8. For API calls (and also accessing the GUI of the wallet) JRS uses port 7876, so if you would like to access your node from other computers (and/or allow people to use your node for API calls) you will also have to open and forward this port.

9. There is one last step to finalize. Go to C:\Users\$USER\AppData\Roaming\JRS\conf and open the file nxt.properties. Add the following:
   - `nxt.allowedBotHosts=127.0.0.1; localhost; [0:0:0:0:0:0:0:1]; *;`
     If you do not want to allow other people to perform http/json API requests on your node, you will have to delete the *, and add the individual IP addresses of specific computers you would like to give access to your node.
   - `nxt.myAddress=X.X.X.X` where X.X.X.X is your public, external IP address.
   - `nxt.adminPassword=setapasswordhere`
     Setting a password is to protect your node. Some API calls for example can only be performed if the password is entered.
   - Optional: if you would like to give your node a specific name (instead of the by default displayed name "Generic Jupiter Node"):
     `nxt.myPlatform=enterthenamehere`

   The above are the minimal required settings for an active node, but currently there is also a node rewards program for people running an active, open API, archival node (more info can be found in these 2 blog posts https://blog.gojupiter.tech/jupiter-node-rewards-program-d1bdcc95a50a https://blog.gojupiter.tech/node-and-forge-rewards-update-2cde4489e8e0). To be eligible for rewards, your nxt.properties file should contain the following parameters:



```
nxt.allowedBotHosts=*
nxt.myAddress=your external IP address
nxt.adminPassword=setapasswordhere
nxt.myPlatform=your JUP address
nxt.apiServerHost=0.0.0.0
nxt.maxPrunableLifetime=-1
nxt.includeExpiredPrunable=true
```

If you want to run your node 24/7 it is advisable to run it on a VPS. This requires some extra steps and can be found in chapter 3.

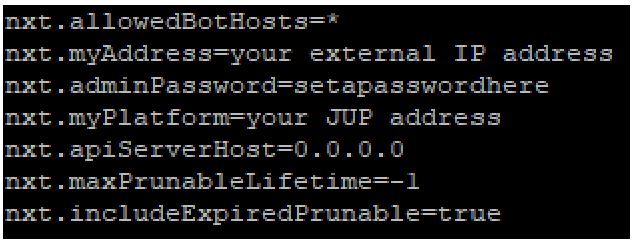
2 Client installation on MacOS

1. To run the JRS client, Java 8 must be installed.
   Direct download link, without login:
   https://javadl.oracle.com/webapps/download/GetFile/1.8.0_261-b12/a4634525489241b9a9e1aa73d9e118e6/unix-i586/jdk-8u261-macosx-x64.dmg
   Or, open Terminal and type:
   ```
   - brew tap AdoptOpenJDK/openjdk
   - brew install --cask adoptopenjdk8
   ```
   Next, you can check your Java version by typing:
   ```
   java -version
   ```
2. Go to https://github.com/jupiter-project/jupiter/releases and download the latest JRS version.
3. Extract the downloaded zip file.
4. To start the client, go into the extracted JRS folder, right-click on start.sh and choose "Open with Terminal". Alternatively, you can open Terminal, navigate to the JRS folder and type:
   ```
   ./start.sh
   ```
   If the client does not launch automatically, open your web browser and go to http://localhost:7876 or https://127.0.0.1:7876 to start it.
5. In (the hidden folder) ~/.jrs/ a folder called nxt_db has been created (if you can't find this folder, go to your home (username) folder and press the keyboard combination cmd + shit + . (dot) to see hidden files and folders). This is the folder where the blockchain database is stored. If you run the client, the blockchain will be downloaded automatically. This may take some time and to speed up this process you can choose to use the snapshot available on the Jupiter toolkit website:
   - Stop the client by right-clicking on the stop.sh file in the JRS folder and choosing "Open with Terminal". Alternatively, open Terminal, navigate into the JRS folder and type:
     ```
     ./stop.sh
     ```
   - Download the mainnet_longnumber.zip file from http://node-eu-de.jupitertoolkit.com/latest_db.php?download.
   - Extract the zip file and move the nxt_db folder to the ~/.jrs/ folder (replace your existing nxt_db folder with the extracted one).
   - Now you can start the wallet.
     Note: when you have not used the wallet for a while and open it, in the bottom right a downloading message appears. This message does not update in real-time. To get rid of the message, log off and log into the wallet again.

Now you are a proud Jupiter node owner.
At the moment you are a "passive" node. This means you connect to the network and can use the wallet, but you are only downloading information. If you are a bit more experienced and especially if you are forging and have your node running 24/7 it would be nice to consider becoming an active peer. This means, besides outgoing connections to download blockchain data from other nodes, you will allow incoming connections so that other nodes can download blockchain data from you. This will increase the strength and security of the network. The only requirement is that you have a static, individual IPv4 address (so if your ISP for example makes use of CGNAT you can't).

6. For incoming peer-to-peer networking requests, JRS uses port 7864. This means your firewall and router should allow this incoming TCP traffic. For API calls (and also accessing the GUI of the wallet) JRS uses port 7876, so if you would like to access your node from other computers (and/or allow people to use your node for API calls) you will also have to open and forward this port. These steps are router specific so just Google "your specific router/modem + port forwarding" if you need any help. Some general info can also be found on websites like https://portforward.com and https://www.noip.com/support/knowledgebase/general-port-forwarding-guide. You can check whether you've configured everything successfully by looking up port 7864 and 7876 for your IP address on https://portchecker.co. It should say that the ports are open.
7. There is one last step to finalize. Go to ~/.jrs/conf/ and create the file nxt.properties. Add the following:
   - `nxt.allowedBotHosts=127.0.0.1; localhost; [0:0:0:0:0:0:0:1]; *;`
     If you do not want to allow other people to perform http/json API requests on your node, you will have to delete the *, and add the individual IP addresses of specific computers you would like to give access to your node.
   - `nxt.myAddress=X.X.X.X` where X.X.X.X is your public, external IP address.
   - `nxt.adminPassword=setapasswordhere`
     Setting a password is to protect your node. Some API calls for example can only be performed if the password is entered.
   - Optional: if you would like to give your node a specific name (instead of the by default displayed name "Generic Jupiter Node"):
     `nxt.myPlatform=enterthenamehere`

   The above are the minimal required settings for an active node, but currently there is also a node rewards program for people running an active, open API, archival node (more info can be found in these 2 blog posts https://blog.gojupiter.tech/jupiter-node-rewards-program-d1bdcc95a50a https://blog.gojupiter.tech/node-and-forge-rewards-update-2cde4489e8e0). To be eligible for rewards, your nxt.properties file should contain the following parameters:

   ```
   nxt.allowedBotHosts=*
   nxt.myAddress=your external IP address
   nxt.adminPassword=setapasswordhere
   nxt.myPlatform=your JUP address
   nxt.apiServerHost=0.0.0.0
   nxt.maxPrunableLifetime=-1
   nxt.includeExpiredPrunable=true
   ```

If you want to run your node 24/7 it is advisable to run it on a VPS. This requires some extra steps and can be found in the next chapter.

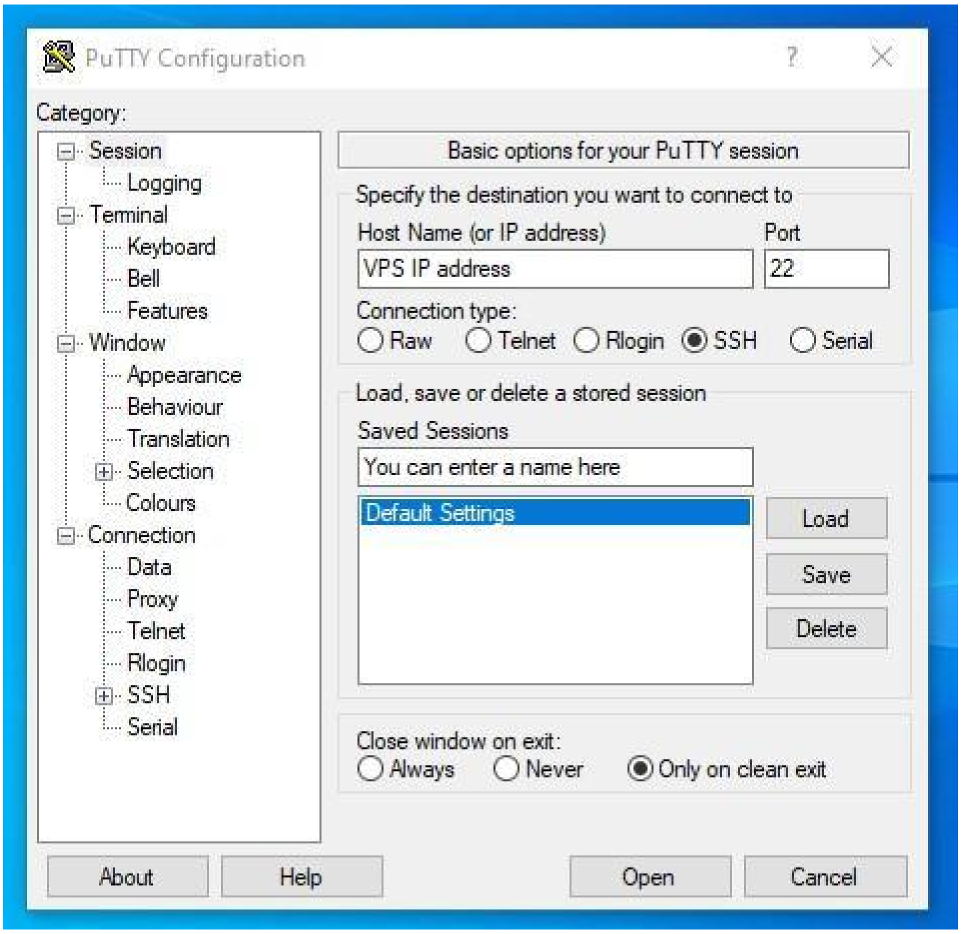
3 Setting up a Jupiter node on Linux VPS

This chapter includes all basic steps for setting up a Linux VPS, installing the JRS software and access it via your Windows computer.
Note: if you are using Linux or a Mac to access your VPS, instead of using Putty to connect, you have to open Terminal and type:

```
ssh user@IP-address
```

where user is the username you use on the VPS (initially most likely `root`) and IP-address is the IP of your VPS. This will connect you to the VPS via SSH, using default SSH port 22.

1. Get yourself a VPS with at least 1 core, 2 GB RAM and 20 GB SSD and a static IP address. Some well-known VPS providers are Digital Ocean, Vultr, OHV and Contabo.
   - Ubuntu 20.04 is my OS of choice.
   - You will receive an email with your login information and IP address.
2. Log into your VPS using Putty.
   - On your Windows computer, download Putty from https://putty.org and install.
   - Open Putty and enter the IP address of your VPS + port 22 (SSH). Next click on the Open button.
     - ▪ If you enter the IP address + port 22 (SSH), add a name in the box below "Saved sessions" and click on Save, next time you want to log into your VPS you will only have to select the name and click on Load to automatically load the IP address and port number.



Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): VPS IP address

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

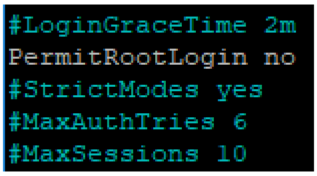
Saved Sessions: You can enter a name here

Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

- A window appears that asks for your login credentials. First enter the username (most likely root) and next the received password (note: you will not see anything when typing the password).
  Now you are in your server.

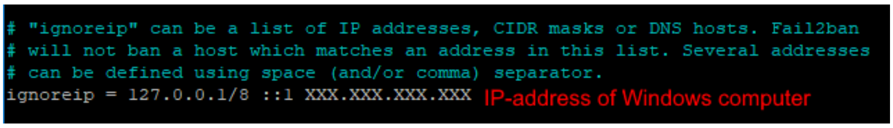
3. Update your OS using the following commands:
   ```
   apt-get update
   apt-get upgrade
   ```
   Note: to copy-paste text from this document to Putty, select text, copy via Crtl+c, then switch to the Putty window and right-click at the prompt in the command line.
4. For security purposes, it is better not to log into your VPS as root user. For this we will create another user via the following command:
   `adduser NEWUSERNAME` (where NEWUSERNAME is the name you want to use).
   Enter all requested information and set a strong password.
5. Now we have a new user account with regular account privileges. However, we may sometimes need to do administrative tasks.
   To avoid having to log out of our normal user and log back in as the root user, we can set up what is known as "superuser" or root privileges for our normal account. This will allow our normal user to run commands with administrative privileges by putting the word "sudo" before each command.
   To add these privileges to our new user, we need to add the new user to the sudo group. By default, on Ubuntu 20.04, users who belong to the sudo group are allowed to use the sudo command.
   As root user, run the following command to add your new user to the sudo group:
   `usermod -aG sudo NEWUSERNAME` (where NEWUSERNAME is the username set in step 4).
6. The next security step is to change the root password you have received by email, by one set by yourself. Simply enter the following command:
   ```
   passwd root
   ```
7. The root user is the user with the most rights on your system. As this account can perform irreversible operations on your server, we recommend you to disable direct root user access via the SSH protocol.
   - Open the configuration file:
     ```
     nano /etc/ssh/sshd_config
     ```
   - Look for "PermitRootLogin".
   - Change "yes" to "no".

     ```
     #LoginGraceTime 2m
     PermitRootLogin no
     #StrictModes yes
     #MaxAuthTries 6
     #MaxSessions 10
     ```
   - Press Crtl+x to exit.
     It now asks to save, type "y".
     Next, it shows File Name to Write: /etc/ssh/sshd_config, press "enter".
   - Restart the service by typing:
     ```
     /etc/init.d/ssh restart
     ```
8. To protect your server against brute force SSH attacks we will install Fail2ban.
   Note: only install Fail2ban if your local Windows computer has a static IP address. This address has to be added to the exclusion list, and this thus won't work if you have a constantly changing IP address.

- `apt-get install -y fail2ban`
- On Ubuntu it starts automatically. You can check whether it's running by typing:
  `sudo fail2ban-client status`
- The file /etc/fail2ban/jail.conf contains the default configuration profile. Updates of the program may result in changes in this file. Therefore, we will not customize our settings in this file, but will create the file fail2ban.local for this. The settings in fail2ban.local will overrule the settings in fail2ban.conf.
  `cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local`
- Open the fail2ban.local file by typing:
  `sudo nano /etc/fail2ban/jail.local`
  At the top, some default properties are shown, there is also a SSH section (the other things are not relevant for us). Change the following settings:

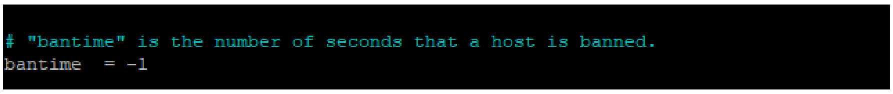
  - [default]
    - `ignoreip`: remove the # and add behind the already configured host IP address, separated with a space, the external IP address of your Windows computer you use to access your server. Now this IP cannot be banned.



```
# "ignoreip" can be a list of IP addresses, CIDR masks or DNS hosts. Fail2ban
# will not ban a host which matches an address in this list. Several addresses
# can be defined using space (and/or comma) separator.
ignoreip = 127.0.0.1/8 ::1 XXX.XXX.XXX.XXX
```

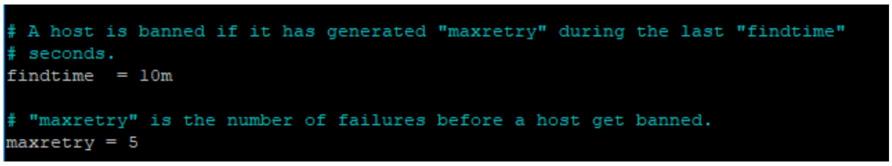
IP-address of Windows computer

    - `bantime`: time an IP address is banned. Default is 10 minutes. Can be changed if you would like. If you would like a permaban, change it into -1.



```
# "bantime" is the number of seconds that a host is banned.
bantime  = -1
```

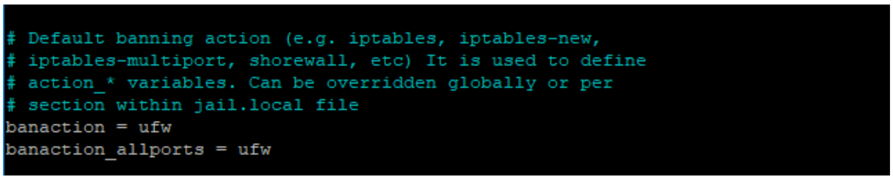
    - `findtime` and `maxretry`: these 2 parameters belong together. If an IP has more than "maxretry" login attempts in "findtime" amount of time, it will be banned. By default, it is 5 attempts in 10 minutes. Can be changed if you would like.



```
# A host is banned if it has generated "maxretry" during the last "findtime"
# seconds.
findtime  = 10m

# "maxretry" is the number of failures before a host get banned.
maxretry = 5
```

    - `Banaction` and `banaction_allports`: as we are going to use the UFW firewall, these must be changed into ufw.



```
# Default banning action (e.g. iptables, iptables-new,
# iptables-multiport, shorewall, etc) It is used to define
# action_* variables. Can be overridden globally or per
# section within jail.local file
banaction = ufw
banaction_allports = ufw
```

  - [sshd]
    It works with default settings (can be adjusted if you want to dig deeper into Fail2ban).
- Press Crtl+x to exit.
  It now asks to save, type "y".
  Next, it shows File Name to Write /etc/fail2ban/jail.local, press "enter".
- Restart Fail2ban:
  `sudo systemctl restart fail2ban`

9. Quit Putty.
10. Open Putty and log in again, this time using your new username and password, set in step 4. At this point, if you would like, you can set up SSH key authentication from your local machine to your Jupiter node. For an overview and instructions, visit https://www.ssh.com/ssh/keygen.
11. Now we are going to set up the UFW firewall to make sure that besides being able to log into the server, only connections to the JRS client are allowed.
    - OpenSSH, the service allowing us to connect to our server, has a profile registered with UFW. You can see this by typing:
      ```
      sudo ufw app list
      ```
    - Enter your password.
    - To make sure the firewall allows SSH connections so that we can log in next time, type:
      ```
      sudo ufw allow OpenSSH
      ```
    - For incoming peer-to-peer networking requests JRS uses port 7864. To allow this, type:
      ```
      sudo ufw allow 7864
      ```
    - You can use your node to access the wallet GUI and perform API requests. For this it uses port 7876. To allow this, type:
      ```
      sudo ufw allow 7876
      ```
    - Now we must enable the firewall, type:
      ```
      sudo ufw enable
      ```
      Type “y” and press “enter”.
    - To see an overview of the allowed traffic, type:
      ```
      sudo ufw status
      ```

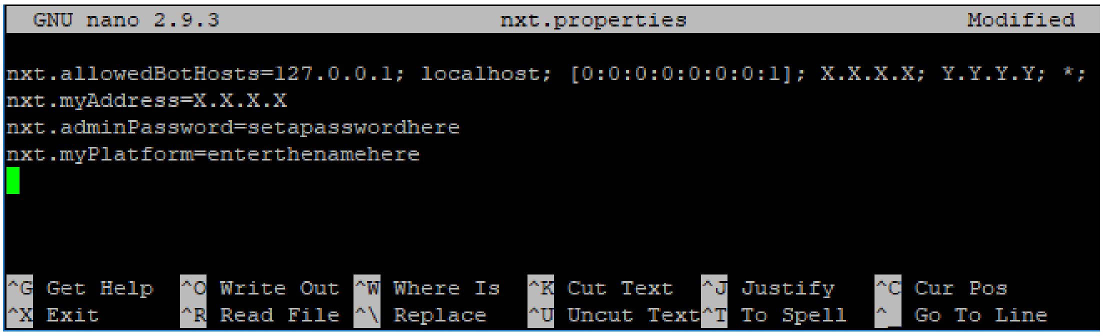
Now we can start with the Jupiter software installation:

12. To run JRS, Java 8 must be installed. This can be done with the following command:
    ```
    sudo apt-get install openjdk-8-jdk
    ```
13. You can check your java version by typing:
    ```
    java -version
    ```
14. Now we need to download the latest JRS version. Surf on your Windows computer to https://github.com/jupiter-project/jupiter/releases and check the link to the latest client (hoover over the client zip file). We need to enter this address at the command line in Putty (this example is based on the info for version 2.4.2, if a newer version has been released, just change the version number):
    ```
    wget
    https://github.com/jupiter-project/jupiter/releases/download/v2.
    4.2/jrs-client-v2.4.2.zip
    ```
15. Install upzip:
    ```
    sudo apt-get install unzip
    ```
16. Next, we need to extract the downloaded zip file:
    ```
    unzip jrs-client-2.4.2.zip
    ```
17. Go into the extracted folder:
    ```
    cd jrs
    ```
18. To get an overview of all files and folders in the jrs folder, type:
    ```
    ls
    ```
19. Go into the conf folder:
    ```
    cd conf
    ```

20. In here, we have to create a file in which we will have to add some additional parameters:

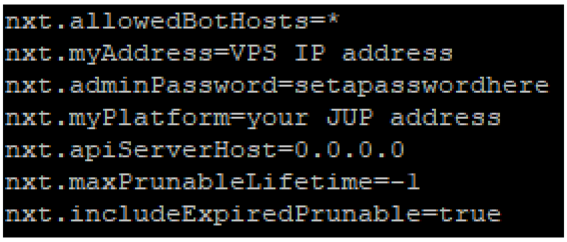
    ```
    nano nxt.properties
    ```

21. Add the following properties:
    - `nxt.allowedBotHosts=127.0.0.1; localhost; [0:0:0:0:0:0:0:1]; X.X.X.X; Y.Y.Y.Y; *;`
      where X.X.X.X. is the IP address of your VPS and Y.Y.Y.Y the public, external IP address of your Windows computer. If you don't want to allow other people to perform http/json API requests on your node, you have to remove *;.
    - `nxt.myAddress=X.X.X.X`
      where X.X.X.X is the IP address of your VPS.
    - `nxt.adminPassword=setapasswordhere`
      Setting a password is to protect your node. Some API calls for example can only be performed if the password is entered.
    - Optional: if you would like to give your node a specific name (instead of the by default displayed name "Generic Jupiter Node"):
      `nxt.myPlatform=enterthenamehere`



```
nxt.allowedBotHosts=127.0.0.1; localhost; [0:0:0:0:0:0:0:1]; X.X.X.X; Y.Y.Y.Y; *;
nxt.myAddress=X.X.X.X
nxt.adminPassword=setapasswordhere
nxt.myPlatform=enterthenamehere
```

The above are the minimal required settings for an active node, but currently there is also a node rewards program for people running an active, open API, archival node (more info can be found in these 2 blog posts
https://blog.gojupiter.tech/jupiter-node-rewards-program-d1bdcc95a50a
https://blog.gojupiter.tech/node-and-forge-rewards-update-2cde4489e8e0). To be eligible for rewards, your nxt.properties file should contain the following parameters:



```
nxt.allowedBotHosts=*
nxt.myAddress=VPS IP address
nxt.adminPassword=setapasswordhere
nxt.myPlatform=your JUP address
nxt.apiServerHost=0.0.0.0
nxt.maxPrunableLifetime=-1
nxt.includeExpiredPrunable=true
```

22. Press Crtl+x to exit.
    It now asks to save, type "y".
    Next, it shows File Name to Write: nxt.properties, press "enter".
23. Go back to the jrs folder, type:
    `cd ..`
24. To start your node, type:
    `bash start.sh`
    Now your node is running! You can close Putty.

25. To access the VPS wallet on your Windows computer, open your browser and go to: `http://X.X.X.X:7876` (where X.X.X.X is the IP address of your VPS).
    And if you would like to perform API calls, the API console can be found at `http://X.X.X.X:7876/test`
    BE CAREFUL, the internet communication with your VPS, so accessing your wallet, goes via an UNSECURED http connection. Therefore NEVER enter the passphrase of the account that holds all your JUP into the wallet you are running on a VPS. Please check chapter 6.2 how to create a safe "2-address setup" (the alternative, adding a domain name to your VPS and set up a SSL certificate lays outside the scope of this manual).
26. The first time, it will take some time before the entire blockchain has been downloaded. While downloading, you are connected to a remote node. For security reasons, please do not enter your seed on this node (you do not know the security precautions this node owner has taken), only use it as read-only.
    Note: the downloading message in the bottom right does not update real-time. To get rid of the message, log off and log into the wallet again.
    To speed up the downloading process, you could download the latest blockchain snapshot from the Jupiter toolkit website.
    - First stop the Jupiter server, by typing the following (being in the jrs folder):
      ```
      bash stop.sh
      ```
    - Download the latest zip file (being in the jrs folder):
      ```
      wget
      http://node-eu-de.jupitertoolkit.com/latest_db.php?download
      ```
      Note: if you copy-paste this line, check whether the "-" between eu-de is still there. If not, you can move to the position using the "left arrow" and add it yourself.
    - Extract the file latest_db.php?download:
      ```
      unzip latest_db.php?download
      ```
      Your existing nxt_db folder (and all files it contains) has to be replaced by the one in the zip file, so answer all three replace questions with "y".
    - Start your node again to download the last blocks:
      ```
      bash start.sh
      ```

4 Client installation on Raspberry Pi

Instructions how to install the Jupiter node/wallet software on a Raspberry Pi can be found over here https://blog.gojupiter.tech/install-jupiter-on-a-raspberry-pi-ac2d93ae8899.

Note 1:

Although the blog post (which has been written a few months ago) tells to install Raspbian, if you would like turn your Raspberry Pi into a Metis container in the future, it is advisable to install 64-bit Ubuntu20.04 as your OS, as Metis isn't compatible with Raspbian (be sure to select 64-bit, 32-bit Ubuntu will have the same problem).

Note 2:

In Part 2 software setup, instead of installing Java, installing unzip, downloading the latest node software and unzip it with the 5 commands displayed, you eventually could choose to install those using an installation script Sigwo has written. However, this script will only work on 64-bit Ubuntu20.04 (it is basically the first iteration of the Metis container script that in addition to the node software will also install Metis web and its dependencies, although Metis web isn't functioning properly at the moment).

Those 5 steps then have to be replaced by the following command:

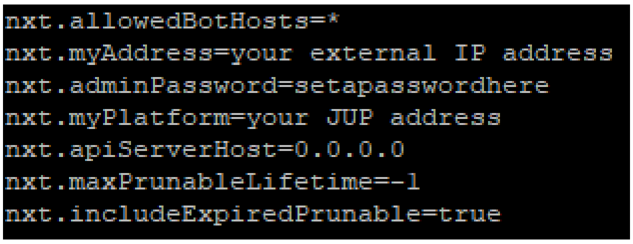
```
curl -o- https://gist.githubusercontent.com/sigwo/85809f9170de9876fbf279d910d791cd/raw/907bd96bbfe94790f77835ddab2ae654d1767f2d/buildMetis.sh | bash
```

After this set up you are running a passive node. This means you connect to the network and can use the wallet, but you are only downloading information. If you are a bit more experienced and especially if you are forging and have your node running 24/7 it would be nice to consider becoming an active peer. This means, besides outgoing connections to download blockchain data from other nodes, you will allow incoming connections so that other nodes can download blockchain data from you. This will increase the strength and security of the network. The only requirement is that you have a static, individual IPv4 address (so if your ISP for example makes use of CGNAT you can't).

1. For incoming peer-to-peer networking requests, JRS uses port 7864. This means your firewall and router should allow this incoming TCP traffic. For API calls (and also accessing the GUI of the wallet) JRS uses port 7876, so if you would like to access your node from other computers (and/or allow people to use your node for API calls) you will also have to open and forward this port. These steps are router specific so just Google "your specific router/modem + port forwarding" if you need any help. Some general info can also be found on websites like https://portforward.com and https://www.noip.com/support/knowledgebase/general-port-forwarding-guide. You can check whether you've configured everything successfully by looking up port 7864 for your IP address on https://portchecker.co. It should say that the port is open.
2. Stop your node by typing the following, being in the jrs folder:
   ```
   bash stop.sh
   ```
3. Go into the conf folder:
   ```
   cd conf
   ```
4. In here, you have to create a file in which you will have to add some additional parameters:
   ```
   nano nxt.properties
   ```

5. Add the following properties:
   - `nxt.allowedBotHosts=127.0.0.1; localhost; [0:0:0:0:0:0:0:1]; *;`
     If you do not want to allow other people to perform http/json API requests on your node, you will have to delete the *, and add the individual IP addresses of specific computers you would like to give access to your node.
   - `nxt.myAddress=X.X.X.X` where X.X.X.X is your public, external IP address.
   - `nxt.adminPassword=setapasswordhere`
     Setting a password is to protect your node. Some API calls for example can only be performed if the password is entered.
   - Optional: if you would like to give your node a specific name (instead of the by default displayed name "Generic Jupiter Node"):
     `nxt.myPlatform=enterthenamehere`

   The above are the minimal required settings for an active node, but currently there is also a node rewards program for people running an active, open API, archival node (more info can be found in these 2 blog posts https://blog.gojupiter.tech/jupiter-node-rewards-program-d1bdcc95a50a https://blog.gojupiter.tech/node-and-forge-rewards-update-2cde4489e8e0). To be eligible for rewards, your nxt.properties file should contain the following parameters:



```
nxt.allowedBotHosts=*
nxt.myAddress=your external IP address
nxt.adminPassword=setapasswordhere
nxt.myPlatform=your JUP address
nxt.apiServerHost=0.0.0.0
nxt.maxPrunableLifetime=-1
nxt.includeExpiredPrunable=true
```

6. Press Crtl+x to exit.
   It now asks to save, type "y".
   Next, it shows File Name to Write: nxt.properties, press "enter".
7. Go back to the jrs folder, type:
   `cd ..`
8. To start your node again, type:
   `bash start.sh`

5 Updating your node

5.1 Windows

Updating your node on Windows is very easy.
If your node is still running, stop it by closing both the wallet and Java windows.

1. Download the latest software package from GitHub https://github.com/jupiter-project/jupiter/releases.
2. Extract the downloaded zip file. Store the extracted folder wherever you want.
3. Go into the folder and double-click on the run.bat file.
   The new version will automatically use the blockchain database and nxt.properties file with your custom settings in the C:\Users\$USER\AppData\Roaming\JRS folder.
4. You can delete the JRS folder containing the old version.

5.2 Linux VPS

1. Log into your VPS using Putty.
2. Go into the jrs folder:
   ```
   cd jrs
   ```
3. Stop your node:
   ```
   bash stop.sh
   ```
4. Go back to your home folder:
   ```
   cd ..
   ```
5. Rename the jrs folder to jrs_old (or other name you want to use):
   ```
   mv jrs jrs_old
   ```
6. Download the latest software package (still being in the home folder). Check https://github.com/jupiter-project/jupiter/releases for the exact link:
   ```
   wget
   https://github.com/jupiter-project/jupiter/releases/download/res
   t_of_the_link
   ```
7. Extract the downloaded zip file:
   ```
   unzip jrs-client-rest_of_the_name.zip
   ```
8. In your home folder you now have a folder called jrs containing the latest software, and a folder called jrs_old containing the previous version (you can check via the command "ls"). In contrary to Windows, where the blockchain database and nxt.properties file are stored in another location, that will directly be accessed by the new version, in Linux, they are stored in the jrs folder itself. This means the newly installed version, just like with your initial installation, does not contain these files yet. So now we are going to copy the folder with database files and the nxt.properties file from the old to the new folder.
   ```
   cp -R jrs_old/nxt_db jrs/
   cp jrs_old/conf/nxt.properties jrs/conf/nxt.properties
   ```
9. Now you can either keep the jrs_old folder as a backup, or delete it via the command:
   ```
   rm -R jrs_old/
   ```
10. Go into the jrs folder
    ```
    cd jrs
    ```

11. Start your node
    `bash start.sh`

6 General wallet info

The Jupiter blockchain is based on the NXT blockchain.
https://nxtdocs.jelurida.com/Getting_started contains a lot of general info. So, if you need some help with the wallet, have a look over there and highly likely you will find your answer.
Just to name some specific pages:

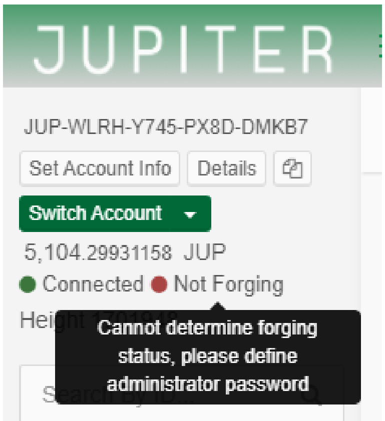
- Account creation: https://nxtdocs.jelurida.com/How_to_create_an_account
- The client interface: https://nxtdocs.jelurida.com/Nxt_client_interface
- Sending JUP: https://nxtdocs.jelurida.com/How_to_send_NXT
- Forging: https://nxtdocs.jelurida.com/Forging_feature
  Leasing: https://nxtdocs.jelurida.com/Account_leasing
- Asset exchange: https://nxtdocs.jelurida.com/Asset_exchange
- API guide: https://nxtdocs.jelurida.com/API

We will look a bit deeper into forging and leasing.

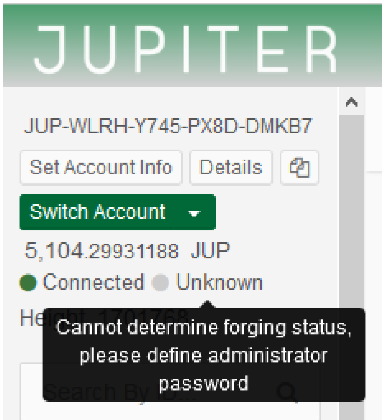
6.1 Forging on your local node

Jupiter is a Proof-of-Stake blockchain and the process of finding new blocks is called forging. The address that finds the block will be rewarded with the transaction fees in it.
To start forging the only things you need are:
- A full node running 24/7.
- A JUP account with more than 1000 JUP that have 1440 confirmations (the more JUP you have, the higher the chance of finding a new block).
- An outgoing transaction so that the public key of your account is known. Easiest way is to send 1 JUP to your own address.

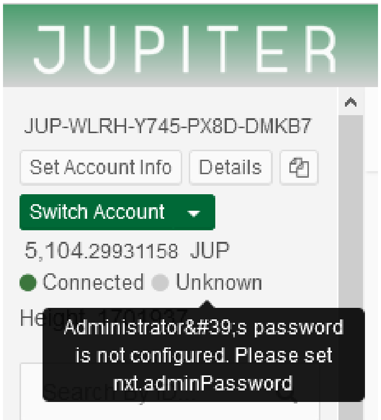
1. Start your local node and open the wallet.
2. You'll now probably see one of these:



JUP-WLRH-Y745-PX8D-DMKB7
5,104.29931158 JUP
● Connected ● Not Forging
Cannot determine forging status, please define administrator password



JUP-WLRH-Y745-PX8D-DMKB7
5,104.29931188 JUP
● Connected ● Unknown
Cannot determine forging status, please define administrator password

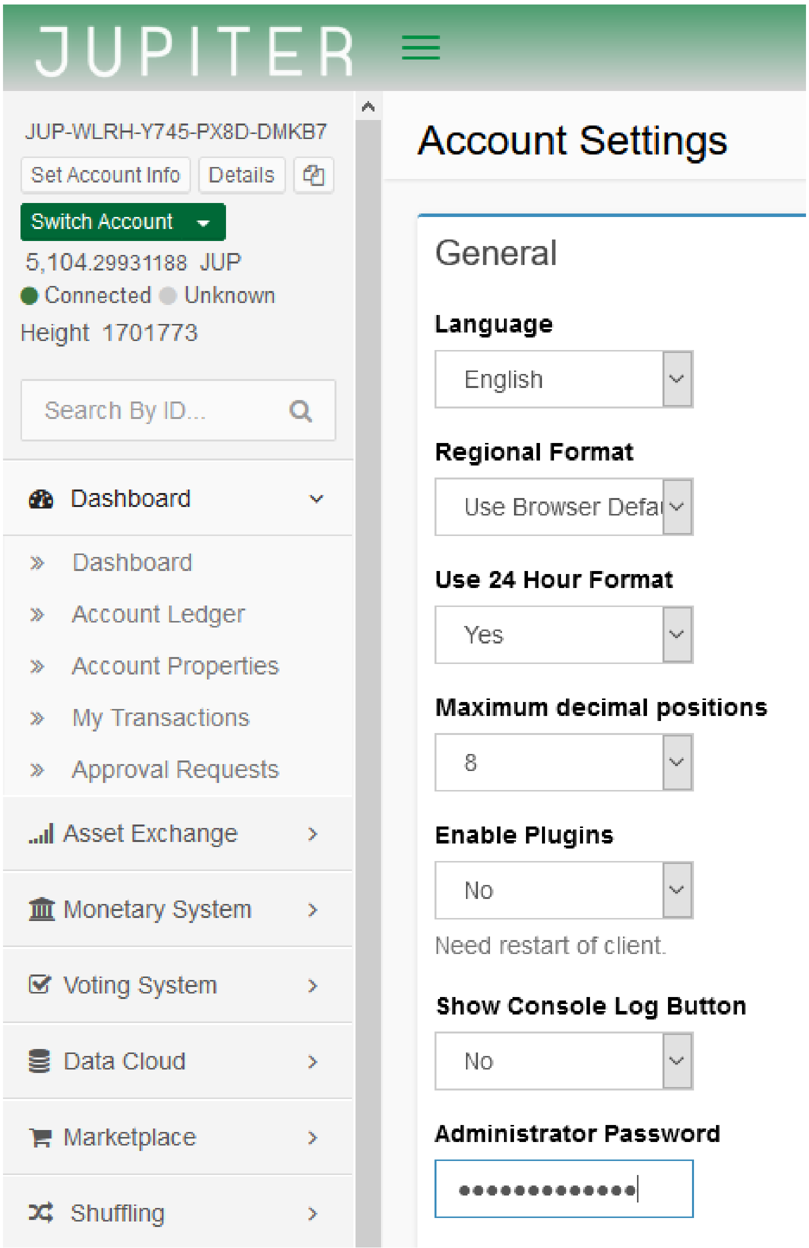


JUP-WLRH-Y745-PX8D-DMKB7
5,104.29931158 JUP
● Connected ● Unknown
Administrator&#39;s password is not configured. Please set nxt.adminPassword

3. Click on Not Forging / Unknown. Now a modal appears. Enter your passphrase and click on the "Start Forging" button. For a moment you see a green bullet with Forging, after which it might turn back to a grey bullet telling Unknown. Although this looks confusing, the account is still forging and there is also a way to make things more clear, see the steps below.

4. If you are running a passive node, you'll first have to set an administrator password in the nxt.properties file.
   - Stop your node by both closing the wallet tab in your browser and Java window.
   - Then have a look at chapter 1, step 9 for Windows, chapter 2, step 7 for Mac and chapter 4, step 2-6 for Raspberry Pi to see where/how to set the admin password.
   - Next start your node and wallet again.

   Active nodes already have configured this.
5. Now you have to add the admin password in the wallet settings. For this, go to Settings (cogwheel) in the top right, select Account Settings. Then in the General section, the bottom field says Administrator Password. Enter here the admin password that you have set in your nxt.properties file.



Account Settings

General

Language: English

Regional Format: Use Browser Defa

Use 24 Hour Format: Yes

Maximum decimal positions: 8

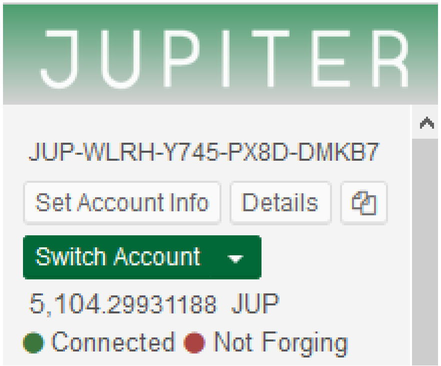
Enable Plugins: No

Need restart of client.

Show Console Log Button: No

Administrator Password: ••••••••••••••

6. If you now click in the grey area around the Unknown bullet/JUP balance/height you'll leave the settings page and you'll see the bullet turns red and says Not Forging.



JUP-WLRH-Y745-PX8D-DMKB7

5,104.29931188 JUP

● Connected ● Not Forging

7. Click on "Not Forging", enter your passphrase and click on the "Start Forging" button. Now the bullet turns green and says you are forging.
   Unless you shut down your node, you keep on forging (so you can close the wallet in your browser, but the Java window has to stay open). If you accidentally stop your node or restart your computer, you will have to start forging again by entering your passphrase.

Additional info regarding forging can be found in this blog post:
https://blog.gojupiter.tech/start-forging-jupiter-9bb0947d8f59

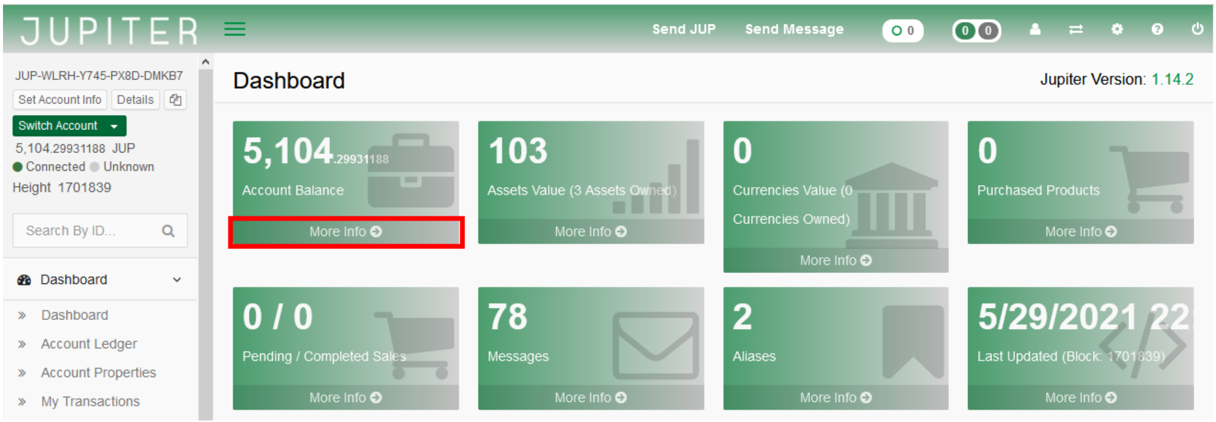
6.2 Forging and leasing on your VPS node

Jupiter is a Proof-of-Stake blockchain and the process of finding new blocks is called forging. The address that finds the block will be rewarded with the transaction fees in it.
To start forging the only things you need are:
- A full node running 24/7.
- A JUP account with more than 1000 JUP that have 1440 confirmations (the more JUP you have, the higher the chance of finding a new block).
- An outgoing transaction so that the public key of your account is known. Easiest way is to send 1 JUP to your own address.

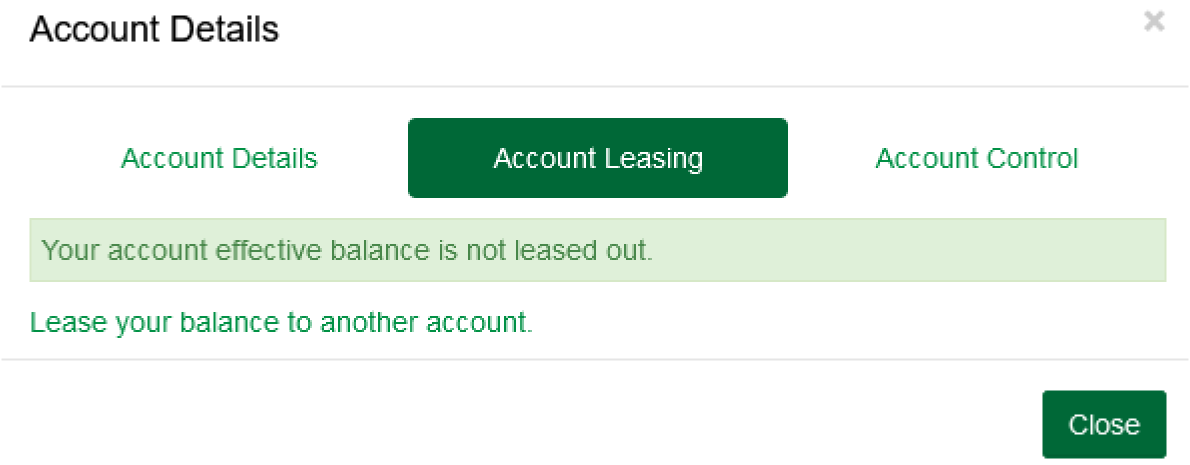
If you are running your node remotely on a VPS, NEVER ENTER THE PASSPHRASE OF THE ACCOUNT THAT IS HOLDING ALL YOUR JUP, even if it is your own node.
Better be safe than sorry, and Jupiter has all the tools on board to prevent security risks, with the so called function "leasing". By leasing, your coins are still in your wallet, you have 100% ownership, but you "rent" them out to an account that is forging, so that the forging power of that account increases. While leasing, you also can still spend the coins in the address.
So this is the ultimate tool to create a two-account setup: a main account securely holding all your coins (that you only use on your local wallet or the team's fully secured online wallet (https://nodes.jup.io)), that leases its coins to your forging account on your VPS.

1. Start the node on your VPS and create a new wallet.
2. Log into your main JUP account using the wallet software on your local computer or the secured web wallet on https://nodes.jup.io.
3. Send a bit over 1000 JUP to the wallet on your remote node and send 1 JUP back to yourself, so that your public key gets known.

4. Now we are going to lease the balance of your main account to the node account. On your dashboard click on More info in the Account Balance block.



5. Select the second tab “Account Leasing”.

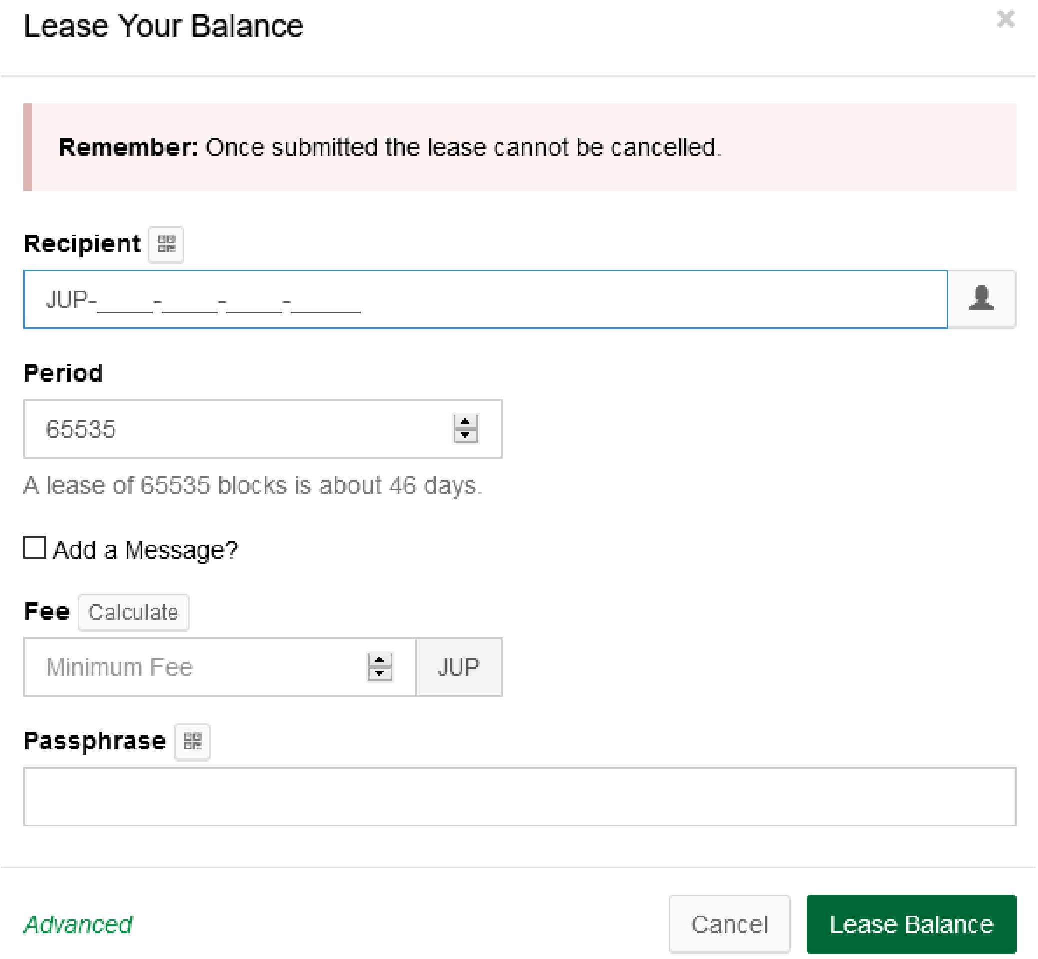


Your account effective balance is not leased out.

Lease your balance to another account.

6. Click on Lease your balance to another account.

7. Enter the JUP address of the account on your node as recipient, leave the period at 65535, enter your passphrase and click on the “Lease Balance” button.



Lease Your Balance

Remember: Once submitted the lease cannot be cancelled.

Recipient

JUP-____-____-____-_____

Period

65535

A lease of 65535 blocks is about 46 days.

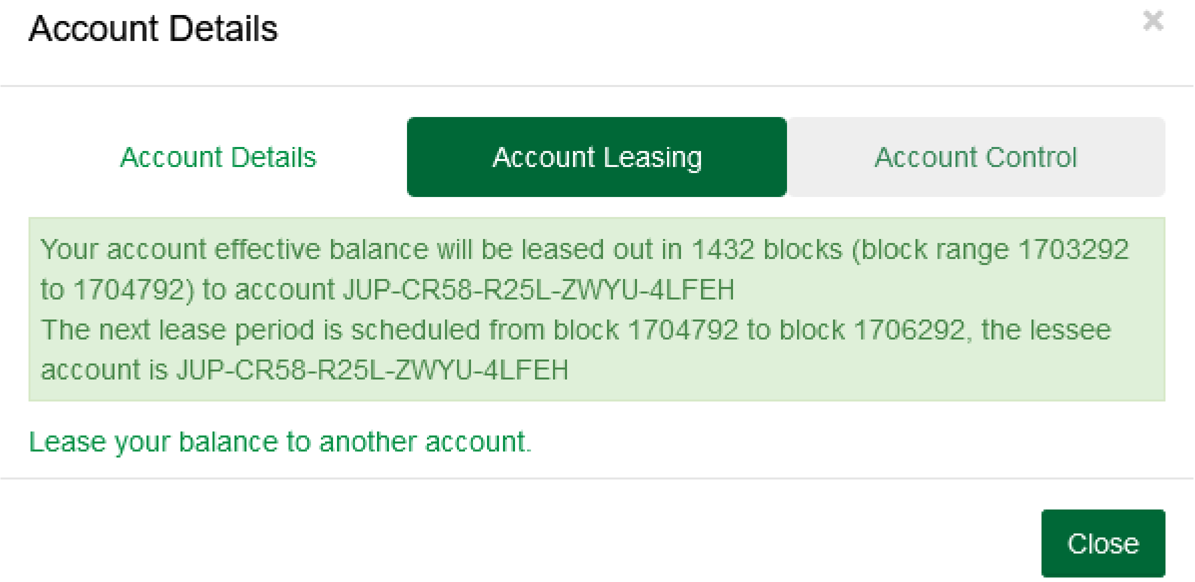
□ Add a Message?

Fee

Minimum Fee JUP

Passphrase

Note that you have to renew the lease every 65k blocks, and every time it takes 1440 blocks before the lease is active. Although, after you have started your lease, you can already perform the transaction for a second period (and as there are >1440 blocks between this transaction and start of the second period, the leasing periods are then adjacent).

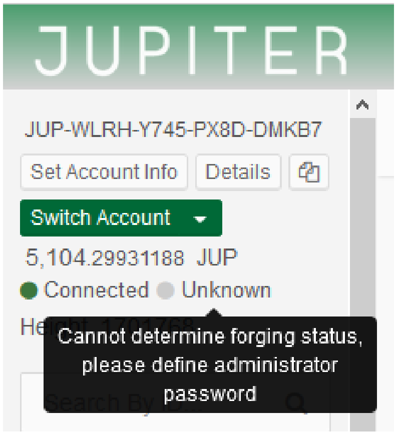


Account Details

Your account effective balance will be leased out in 1432 blocks (block range 1703292 to 1704792) to account JUP-CR58-R25L-ZWYU-4LFEH
The next lease period is scheduled from block 1704792 to block 1706292, the lessee account is JUP-CR58-R25L-ZWYU-4LFEH

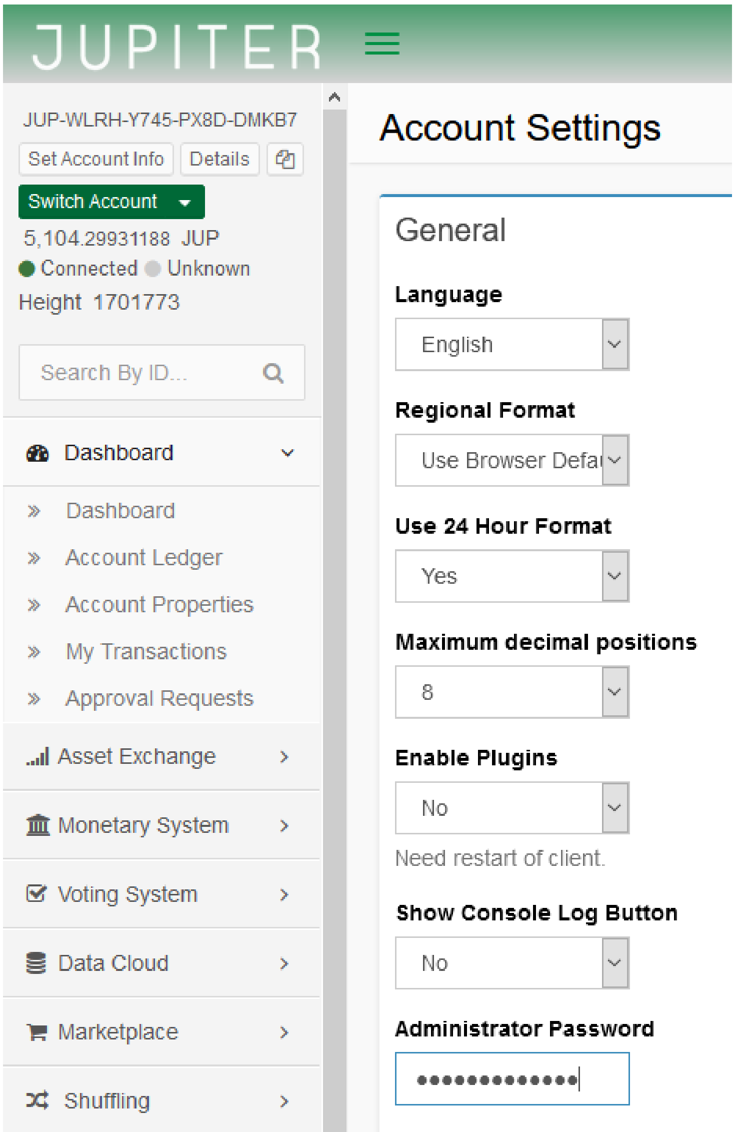
Lease your balance to another account.

8. After the >1000 JUP on your VPS JUP account have >1440 confirmations, you can start forging. Open the wallet on your VPS and log in with your VPS JUP account.
9. The forging bullet probably says Unknown. To fix this, you'll have to add the admin password in the wallet settings.



Cannot determine forging status, please define administrator password

For this, go to Settings (cogwheel) in the top right, select Account Settings. Then in the General section, the bottom field says Administrator Password. Enter here the admin password that you have set in your nxt.properties file.



JUP-WLRH-Y745-PX8D-DMKB7

5,104.29931188 JUP

● Connected ● Unknown

Height 1701773

Account Settings

General

Language: English

Regional Format: Use Browser Defa

Use 24 Hour Format: Yes

Maximum decimal positions: 8

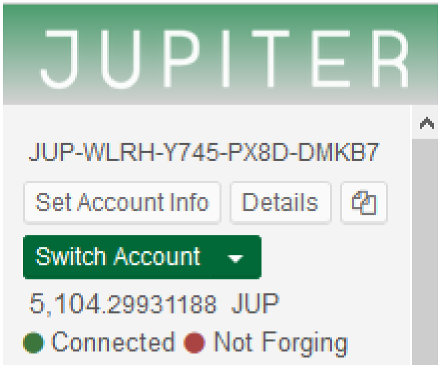
Enable Plugins: No

Need restart of client.

Show Console Log Button: No

Administrator Password: •••••••••••••

10. If you now click in the grey area around the Unknown bullet/JUP balance/height, you'll leave the settings page and you'll see the bullet turns red, saying Not Forging.



JUP-WLRH-Y745-PX8D-DMKB7

5,104.29931188 JUP

● Connected ● Not Forging

11. Click on "Not Forging", enter your passphrase and click on the "Start Forging" button. Now the bullet turns green and says you are forging.
Unless you shut down your node, you keep on forging (so you can close the wallet in your browser). If you stop your node on the VPS, you will have to start forging again by entering your passphrase.

Additional info regarding forging can be found in this blog post:
https://blog.gojupiter.tech/start-forging-jupiter-9bb0947d8f59

7 Mainnet JUP, ERC20 JUP and BEP20 JUP

While mainnet JUP is needed to use dApps on the Jupiter blockchain, there are 2 more versions available, ERC20 JUP and BEP20 JUP, that are used for trading. You can switch 1:1 between the versions via https://swap.jup.io.
As a lot of info is updated now and then, instead of writing things down extensively with the risk the links might get outdated, below will only follow a short description of the different tokens and for more info you can have a look at the following link, that contains an up-to-date list of all important links related to Jupiter:
https://docs.google.com/document/d/1iziiSDnbaOBIkbAJyzQFB8MRNjo8D7k8WxahwEKMgWM

Mainnet JUP:

- Can be stored in the native Jupiter wallet (web or standalone).
- Can be used to create and use Jupiter dApps.
- You can set up your own node and start forging to collect a share of the transaction fees.
- If you don't want to run a node yourself, you can still earn transactions fees by participating in the FORGE pool.

ERC20 JUP:

- Contract address: 0x4B1E80cAC91e2216EEb63e29B957eB91Ae9C2Be8
- Can be stored in any wallet that supports ERC20 tokens, for example Myetherwallet, MyCrypto, Trustwallet and Metamask (also in combination with Ledger and Trezor).
- Is used by exchanges, like Kucoin and Stex.
- Can be traded on DEXes like Uniswap.

BEP20 JUP:

- Contract address: 0x0231f91e02debd20345ae8ab7d71a41f8e140ce7
- Can be stored in any wallet that supports BEP20 tokens, for example Trustwallet and Metamask (also in combination with Ledger and Trezor).
- Is used by exchanges, like Kucoin.
- Can be traded on DEXes like Pancakeswap.

8 Sources used to create this manual

- https://github.com/jupiter-project/jupiter
- https://github.com/jupiter-project/jupiter/releases
- https://nxtdocs.jelurida.com/Set_up_a_public_node_on_a_VPS
- https://nxtdocs.jelurida.com/Getting_started
- https://www.wavesassist.com/installation-waves-node
- https://www.digitalocean.com/community/tutorials/initial-server-setup-with-ubuntu-18-04
- https://www.digitalocean.com/community/tutorials/how-to-protect-ssh-with-fail2ban-on-ubuntu-14-04
- https://www.linode.com/docs/security/using-fail2ban-to-secure-your-server-a-tutorial
- https://www.ssh.com/ssh/keygen
- https://jupitertoolkit.com