Blockchain & Cryptocurrencies

Komodo software Installation Guide

Vaibhav Murkute (vaibhav.murkute@mavs.uta.edu)

CSE@UTA

Requirements:

- **Linux (any Debian-based distribution)**: Installation steps for OS X, Windows and Ubuntu machines, are mentioned on the website: https://docs.komodoplatform.com/komodo/install-Komodo-manually.html#installing-komodo-on-osx . For Windows or OS X installation you can follow the steps mentioned on this website. But I would strongly recommend using Ubuntu 16 or 18, running natively or on a VM, since there are some open issues in OS X installation, as I am writing this document.
- **8 GB RAM with 64-bit processor**: 4 GB RAM would work too, but it takes a lot of time to install this software. Also it freezes the computer during installation and fails sometimes requiring you to start all over again. Installation on 4 GB RAM machine takes around 4-5 hrs to complete. Just an FYI!

Assuming you are running Ubuntu (or any Debian-based Linux distribution), open up a terminal and follow these steps:

1. Update all the installed packages on your system:

- sudo apt-get update
- sudo apt-get upgrade -y

2. Install these dependency packages: That's a one-line command, so copy the entire thing and run it at once.

- sudo apt-get install build-essential pkg-config libc6-dev m4 g++-multilib autoconf libtool ncurses-dev unzip git python zlib1g-dev wget bsdmainutils automake libboost-all-dev libssl-dev libprotobuf-dev protobuf-compiler libgtest-dev libqt4-dev libqrencode-dev libdb++-dev ntp ntpdate software-properties-common curl libcurl4-gnutls-dev cmake clang libsodium-dev -y

3. Navigate to your home directory and install Nanomsg. Run each command (line) here individually:

- cd ~

- git clone https://github.com/nanomsg/nanomsg
- cd nanomsg
- cmake . -DNN_TESTS=OFF -DNN_ENABLE_DOC=OFF
- make -j2
- sudo make install
- sudo ldconfig

If you see any error saying “libgmp headers missing”, ignore it!

4. Again Navigate to home directory and fetch Komodo repository from git:

- cd ~
- git clone https://github.com/jl777/komodo

5. Switch to “komodo” directory and checkout ‘dev’ branch. (so that you don’t mess around with Master!)

- cd komodo
- git checkout dev

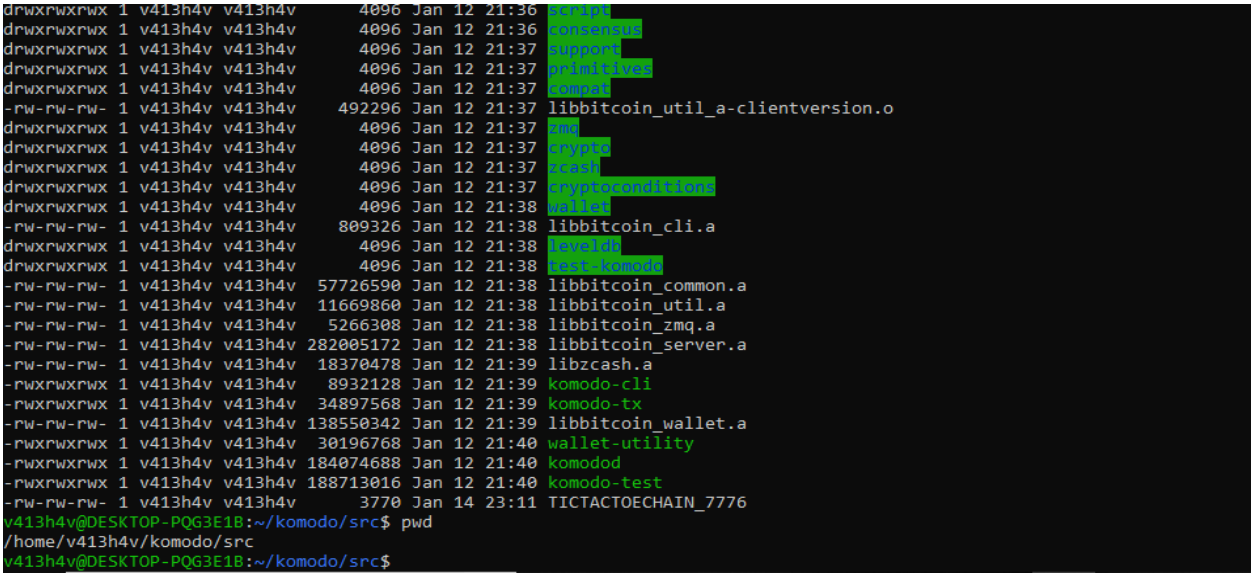
6. Now this step is REALLY Important. Assuming you are still in ‘komodo’ directory, run these 2 commands one-by-one:

- ./zcutil/fetch-params.sh
- ./zcutil/build.sh -j$(nproc)

   First line will take some time. Once the first one is successful, execute the second line. Here you can change the variable **$(nproc)** to any number you want. This number defines the number of threads to deploy for this installation. $nproc will give a number based on your hardware. Keep it as it is. Or to speed up the process, you can set it to ‘8’, provided your hardware can support 8 threads at a time. So the new command will look like: `./zcutil/build.sh -j8`

Second line (command) will take a lot of time, based on your hardware specs. May be an hour or two. So sit back, grab a cup of coffee and pray that this step won’t fail!

If everything till now works out fine, you should see these files in your “komodo/src/” directory:



Focus only on these 2 files:

- komodod
- komodo-cli

We will use these 2 files for all our blockchain operations. If you don’t see these 2 files in komodo/src/ directory, blame your stars and re-run step 6. It will re-build your project.

This 6th step often fails on a 4 GB RAM machine. I had to try 6 times to get this step right on my 4 GB RAM, intel-i3-processor komodo node. So the bottom line is: Keep trying step 6 until you see those 2 files in /komodo/src/ directory and don’t take NO for an answer!

(*If you still can’t get this step right, don’t worry, let me know, we will figure something out.)

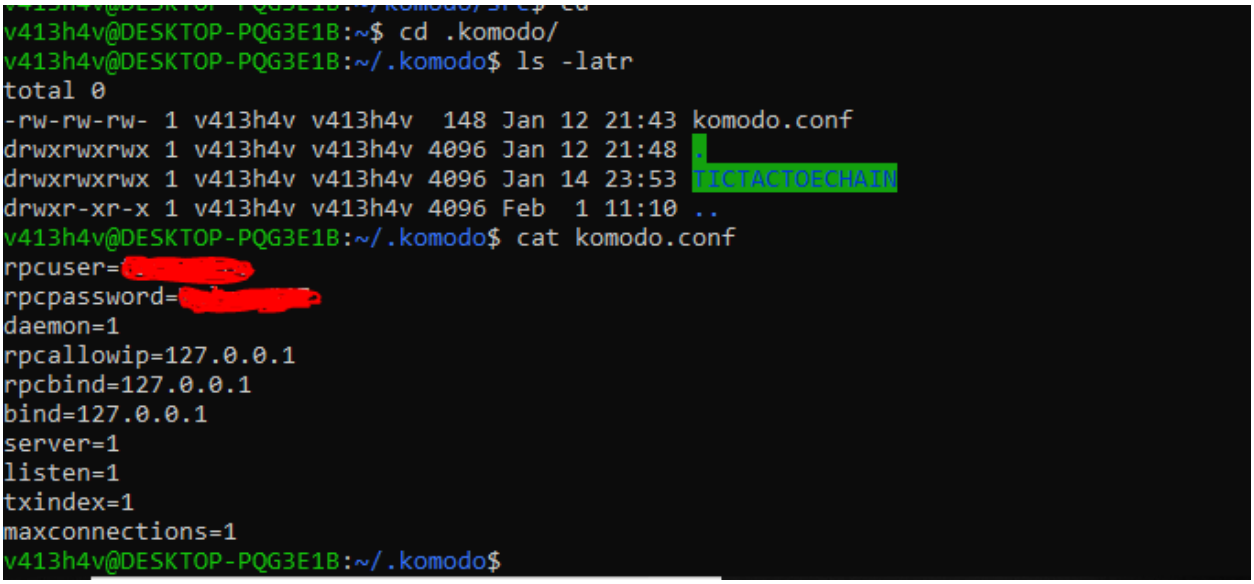
7. If you’ve reached this far, congrats! Your prayers have been answered and you are just one step away from completing this installation. Navigate to the home directory, create a folder named “.komodo” (mind the dot(.) in front) and inside that folder create “komodo.conf” file:

- cd ~
- mkdir .komodo
- cd .komodo
- nano komodo.conf

You can use nano or other editor of your choice (like Vim) to edit the komodo.conf file.

Add below content into the komodo.conf file.

rpcuser=your_user_name
rpcpassword=your_password
daemon=1
rpcallowip=127.0.0.1
rpcbind=127.0.0.1
bind=127.0.0.1
server=1
listen=1
txindex=1
maxconnections=1



Note that "**maxconnection**" here is set to 1 because the blockchain I created had only two nodes in it (i.e. one more node to connect to …. i.e. just one connection…. i.e maxconnections=1….you got it Sherlock!). This parameter can be set to any number when we scale our project to support multiple nodes.

Also, please note that the same configuration file is available on: https://docs.komodoplatform.com/komodo/install-Komodo-manually.html <---- but do not add 'addnode' parameters into YOUR komodo.conf file, as the website says. (Unless you want to add your device to Komodo's own blockchain). Komodo's own blockchain has the height of 1225845, as I am writing this document. That means if you add those nodes, as mentioned in the website's komodo.conf file and once you start your komodo software, it will start syncing up your komodo wallet with other chained nodes and Trust me, you wouldn't wanna do that because syncing 12,00000+ blocks takes more than 24-hrs to complete. So avoid doing this. Just stick to the komodo.conf given in this document.

And that's the end of the installation part.

Komodo Software Usage:

This komodo software can be used to operate on Komodo's own blockchain (Komodo-Coin) and also to create your own customized asset-chain (just another fancy name for blockchain). In our case, we are creating our own blockchain, so let's just focus on the latter part.

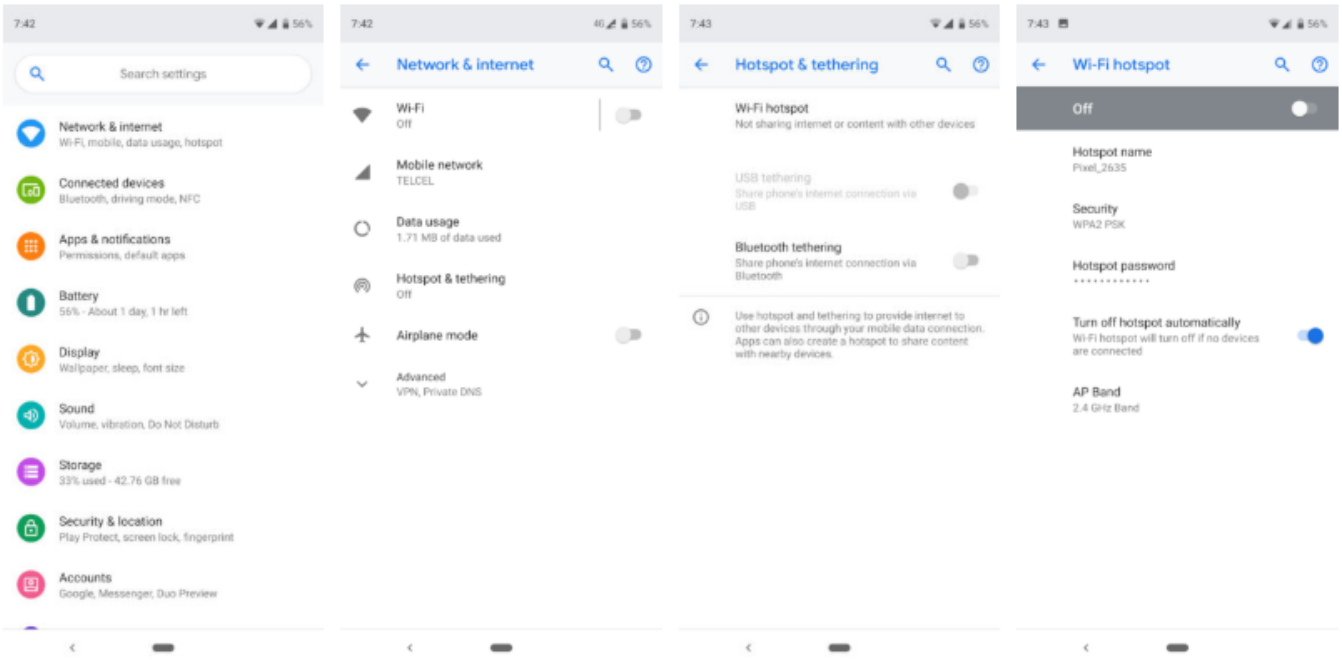
The 2 important files I mentioned before:

- **komodod:** This executable file starts the komodo server on your machine and communicates with other blockchain nodes. So this file needs to be running all the time while you plan to use your blockchain.
- **Komodo-cli:** This executable file is used to run commands on your blockchain. Basically what it does is, it interacts with your already running komodod, passes your commands to it and fetches the result. So all the commands available in this API are supposed to be run with this executable.

Please note that Komodo blockchain (or any blockchain for that matter) needs to have atleast 2 connected nodes (cuz that's when we can call it a "chain" LOL). If you have one more spare computer with you, install komodo in that machine too. (Follow the same steps mentioned above). For second node you can try using Amazon's AWS cloud instance. You can also try setting up another virtual machine on the same computer. (but rumor has it that creating a second node on a VM doesn't work with Komodo! You can still try and please let me know if it works.) I would recommend using your another spare computer or AWS instance.

If you can't manage to add second node, don't worry. We can still work with just one node and play around with all komodo commands. I will show you how in later part.

You can create your own network using a router or a mobile hotspot. For Android phones,



Go to ‘Settings’ → ‘Network & Internet’ → ‘Hotspot & Tethering’ → Turn on WiFi hotspot

Now connect all the nodes you have to this WiFi hotspot to get them all under one subnet. You can now use all these connected nodes to extend your asset-chain.

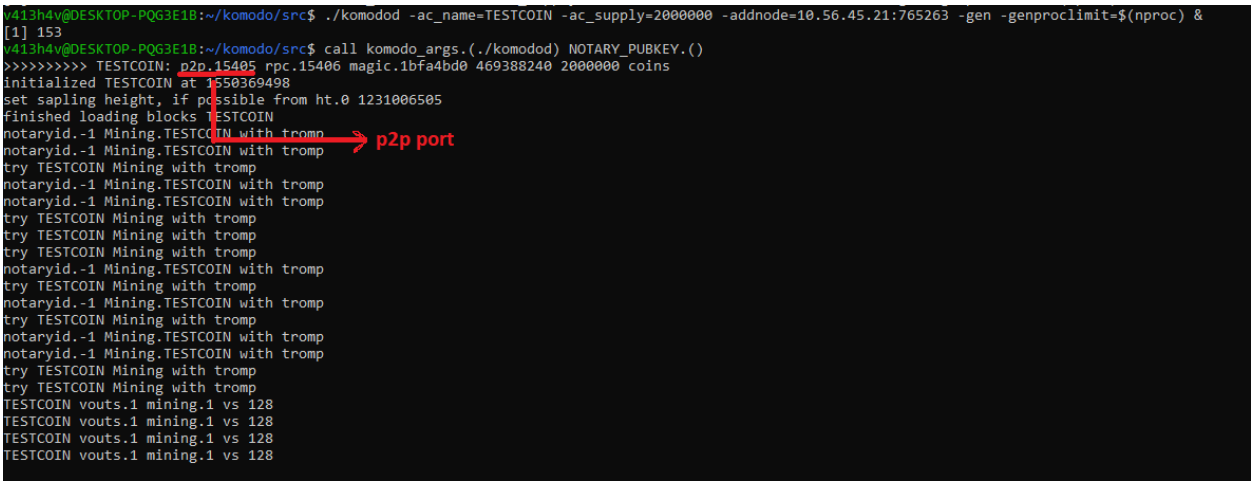
Here is how you create your own asset-chain:

Just open up a terminal, navigate to ~/komodo/src/ and type following command:

- ./komodod -ac_name=**YOUR_ASSET_CHAIN_NAME** -ac_supply=2000000 -addnode=**IP_ADDRESS_OF_SECOND_NODE**:**P2P_PORT_ON_SECOND_NODE** &

This entire line is one command. So run entire line at once. Your asset chain name could be anything, just try to use Capital letters (without space or hyphen etc). ← That’s just a standard suggestion.

Note: All the commands shown in official Komodo API guide (online), do not have ‘-ac_name’ parameter in them. That is because those commands are written for Komodo’s own chain (Komodo-Coin). If you want to run these commands for your asset-chain, you will have to include this ‘-ac_name’ parameter in all ./komodod commands. Failing to do so might raise an error: **couldn't connect to server: unknown (code -1),** if we do not have Komodo-Chain installed on our system.



You can find the P2P port of second machine once you start komodod on the second machine.

If you do not have second node, just put any random IP and port for ‘**addnode’** parameter. It will still start the komodod, but only with one node, which is itself.

Don’t forget the colon (:) between IP and the port number. The “**ac_supply**” parameter defines the number of initial pre-mined (available) coins for that asset chain. ←genesis Block. You can set this cap to any number.

But to own these coins, the second node (or any other node) will have to mine those coins. Whoever mines the genesis block, owns the entire initial ac_supply coin cap.

On second node run the same command with same -ac_name and the IP address of first node. To enable mining on the second node just append "**-gen -genproclimit=$(nproc)**" in the previous command:

- ./komodod -ac_name=**YOUR_ASSET_CHAIN_NAME** -ac_supply=2000000 -addnode=**IP_ADDRESS_OF_FIRST_NODE**:**P2P_PORT_ON_FIRST_NODE** -gen -genproclimit=$(nproc) &

Note: *If you have only one node, then that one node has to mine the genesis block to own those coins. So while creating your blockchain append "-gen -genproclimit=$(nproc)" parameter too, so that this node can start mining the genesis block.* ***Again, without "-gen -genproclimit=$(nproc)" ←this parameter, no komodo node will start mining.*** *(no mining, no coins).*

- **Each asset-chain will have it's own wallet. To get your "wallet.dat" file:**
  - **Navigate to ~/.komodo/**
  - **In this directory, you will see a folder named same as your Asset chain's name. This folder includes all the metadata related to your asset chain:**
    - **Wallet.dat: wallet file for your asset chain**
    - **Debug.log: log file for your asset chain. ← take a look at it to understand how this asset chain actually works.**
    - **YOUR-ASSET-CHAIN.conf : Configuration file for your asset chain**

```
v413h4v@DESKTOP-PQG3E1B:~/.komodo/TICTACTOECHAIN$ ls -ltr
total 224
-rw-rw-rw- 1 v413h4v v413h4v   177 Jan 12 21:48 TICTACTOECHAIN.conf
-rw------- 1 v413h4v v413h4v     0 Jan 12 21:48 db.log
drwx------ 1 v413h4v v413h4v  4096 Jan 12 21:48 blocks
-rw------- 1 v413h4v v413h4v  1729 Jan 14 00:30 komodostate
-rw------- 1 v413h4v v413h4v 90112 Jan 14 23:53 wallet.dat
-rw------- 1 v413h4v v413h4v  5106 Jan 14 23:53 peers.dat
-rw------- 1 v413h4v v413h4v 33910 Jan 14 23:53 fee_estimates.dat
-rw------- 1 v413h4v v413h4v     3 Feb 12 14:20 komodod.pid
drwx------ 1 v413h4v v413h4v  4096 Feb 12 14:20 notarisations
drwx------ 1 v413h4v v413h4v  4096 Feb 12 14:20 chainstate
-rw------- 1 v413h4v v413h4v   540 Feb 12 14:20 komodostate.ind
drwx------ 1 v413h4v v413h4v  4096 Feb 12 14:20 database
-rw------- 1 v413h4v v413h4v 83299 Feb 12 14:22 debug.log
```

Now that you have created your asset chain, it's time to move on to the real action. Now you can use all komodo API commands with 'komodo-cli' tool to operate on your asset chain.

To start using komodo-cli, fire up a new terminal. (do not stop the already running **komodod** instance.). Navigate to ~/komodo/src/ and type-in command in following format:

All your commands will follow this syntax:

- **./komodo-cli -ac_name=YOUR_ASSET_CHAIN_NAME <command>**

You can find the whole list of available command by running:

- **./komodo-cli -ac_name=YOUR_ASSET_CHAIN_NAME --help**

Or you can visit: https://developers.komodoplatform.com/basic-docs/komodo-api/

Listed below are most the frequently used, interesting commands, HAND-PICKED (!) just for you, so that you don't have to go through entire API. But I would highly recommend checking out other commands too, on that website.

- **addnode "node" "add|remove|onetry"** : attempts to add or remove a node from the addnode list, or to make a single attempt to connect to a node.

  Example: ./komodo-cli -ac_name=TICTACTOECHAIN addnode "192.168.0.6:8233" "onetry"

- **getnetworkinfo**: returns an object containing various state info regarding P2P networking.

  Example: ./komodo-cli -ac_name=TICTACTOECHAIN getnetworkinfo

- **getblock hash|height**: returns the block's relevant state information.

  Example: ./komodo-cli -ac_name=TICTACTOECHAIN getblock 2

- **getblockchaininfo:** returns a json object containing state information about blockchain processing.

  Example: ./komodo-cli -ac_name=TICTACTOECHAIN getblockchaininfo

- **getaddednodeinfo dns ("node"):** returns information about the given added node, or all added nodes.If dns is set to false, only a list of added nodes is returned. Otherwise, connection information is also provided.
  Example: ./komodo-cli -ac_name=TICTACTOECHAIN getaddednodeinfo true
  ./komodo-cli getaddednodeinfo true "78.47.205.239"

- **validateaddress "address":** returns information about the given address.
  Example: ./komodo-cli -ac_name=TICTACTOECHAIN validateaddress "RGzE6JsF5nFQfEFZeAzedf3zkKhTvaDhPT"

- **gettransaction "txid" :** queries detailed information about transaction txid. This command applies only to txid's that are in the user's local wallet.

  Example: ./komodo-cli -ac_name=TICTACTOECHAIN gettransaction "4a5e1e4baab89f3a32518a88c31bc87f618f76673e2cc77ab2127b7afdeda33b"

- **getreceivedbyaddress "address"** minconf **:** returns the total amount received by the given address in transactions with at least minconf confirmations. minconf is optional.

  Example: ./komodo-cli -ac_name=TICTACTOECHAIN getreceivedbyaddress "RGzE6JsF5nFQfEFZeAzedf3zkKhTvaDhPT"

- **sendtoaddress "address" amount ("comment" "comment-to" subtractfeefromamount)**: sends an amount to a given address. The amount is real and is rounded to the nearest 0.00000001

  Example: ./komodo-cli -ac_name=TICTACTOECHAIN sendtoaddress "RBtNBJjWKVKPFG4To5Yce9TWWmc2AenzfZ" 0.1

- **getwalletinfo**: The getwalletinfo method returns an object containing various information about the wallet state

  Example: ./komodo-cli -ac_name=TICTACTOECHAIN getwalletinfo

- **listaddressgrouppings**: returns an object containing all your addresses and their balances.

  Example: ./komodo-cli -ac_name=TICTACTOECHAIN listaddressgroupings

- **getnewaddress ("account")**: The getnewaddress method returns a new address for receiving payments.

  Example: ./komodod -ac_name=<YOUR_ASSET_CHAIN_NAME> getnewaddress

- You can stop komodod with: ./komodo-cli -ac_name=TICTACTOECHAIN stop

For more interesting commands, refer the komodo-API website mentioned above.

If your blockchain has only one node, you can't really transfer coins to someone else. To test the coin-transfer (or to generate transactions), you can send coins to yourself with "**sendtoaddress**" command. You can generate more addresses by "**getnewaddress**" command and send coins to those new addresses.

Once you send coins with sendtoaddress, it will return the 'transaction ID'. To check that transaction, you can use "**gettransaction**" command. Keep an eye on '**debug.log'** file to see what is actually going on in the background.

To check coin balance for each address, use command: "**listaddressgroupings**".

Sooo, that’s pretty much it! Checkout other commands, build something interesting, go crazy!

I hope you enjoy it as much as I enjoyed writing this document! If you have any question or if you are stuck at any step, please feel free to contact me at: vaibhav.murkute@mavs.uta.edu.

Peace.