Key Generation

Important information

(i) Grandpa key will be in the ed25519 scheme
(ii) All other keys will be in the sr25519 scheme, it is the default scheme for the "subkey generate" command.
(iii) I recommend first generating all 5 Granpa keys and then generating all other keys

Steps

Step 1: Generate keys using the subkey.
*Note: If you have built the substrate-node-template, "**subkey**" is also installed. To confirm check in the terminal with this command - **"subkey --version".** If Subkey is not installed follow the steps to install "**subkey**" from here: https://docs.substrate.io/v3/tools/subkey/*
To generate the Grandpa key please use this command -
subkey generate --scheme ed25519

Please generate all the other keys using the command -
subkey generate

This command will generate the data like this example -
Secret phrase runway item best include medal subway when famous similar depth spend boring account:
Secret seed:
0x33b276a749d163368e1ea44d9855cee412a966d1d25abd8127039d8506e94bc6
Public key (hex):
0x00cc8561d64446934d1de9c031403c666a04af71707518c914cea2bde365dd33
Account ID:
0x00cc8561d64446934d1de9c031403c666a04af71707518c914cea2bde365dd33
SS58 Address: 5C5kcV2HybAQp1hbMbqwefjQ6LNY1jaBnr1dZD7Fn3QHGzx1

Step 2: Please make a similar document to store this data for all 30 keys and keep this document in a secure place. For example -
Validator 1:

Validator 1 Stash Account:

Secret phrase runway item best include medal subway when famous similar depth spend boring account:
Secret seed:
0x33b276a749d163368e1ea44d9855cee412a966d1d25abd8127039d8506e94bc6
Public key (hex):
0x00cc8561d64446934d1de9c031403c666a04af71707518c914cea2bde365dd33

Account ID:
0x00cc8561d64446934d1de9c031403c666a04af71707518c914cea2bde365dd33

Validator 1 Controller Account:

Secret phrase `brave swarm own erupt often electric power train online special interest law` is account:
Secret seed:
0xd7453420062f3c90ad5851d591d1c8a8b00803ab17cc66fb50e598d93a1b1ec5
Public key (hex):
0x8642e4f26dff4245950325ae7356ce8439c31c03ddeb7b0fd05983b9cf0c155a
Account ID:
0x8642e4f26dff4245950325ae7356ce8439c31c03ddeb7b0fd05983b9cf0c155a
SS58 Address: 5F6k9Pdjwt5zaoKtE5S5AJLrhPAxextb8TJeBJ38Q21Rjqou

Validator 1 Babe Session Key:

Secret phrase `concert cycle situate problem strike seek blind follow fluid claim despair maze` is account:
Secret seed:
0x1844207407762ba01bcc347f5e1df318eaaad9ad0897b161073a35c20b4c8650
Public key (hex):
0x90141672877953eb30fe9ce4d5bf7a63ce6b3261b058bdebce0a843b467c8c5c
Account ID:
0x90141672877953eb30fe9ce4d5bf7a63ce6b3261b058bdebce0a843b467c8c5c
SS58 Address: 5FKcihpPuHs4k4YDNYzb9f4SnpaMfJGxFE4TVFPPXsuZvQsF

Validator 1 Grandpa Session Key:

Secret phrase `sample economy fatal unveil bracket behind leader guide dwarf negative follow write` is account:
Secret seed:
0xc097e1861e7bc42f44d229d71f2b0fb9e55e02a3a7ba98d2fe114e6a86828f2d
Public key (hex):
0xc3738422c4ac5a3a4f47c63739449460d98d34edd9ccd88f7b92aa4f3afa9347
Account ID:
0xc3738422c4ac5a3a4f47c63739449460d98d34edd9ccd88f7b92aa4f3afa9347
SS58 Address: 5GUyWYNpFivV9g7AZxunUGDjavKTFpPgHgufSAd423YU6fJn

...

Validator 2:

Validator 2 Stash Account:

Secret phrase `cup brass tent museum famous bird vendor hammer sea elder auto know` is account:
  Secret seed:
0x02fb2318a5899d46d73e81d786a40554e31aea3eb775ed9f60785dd38406c06b
  Public key (hex):
0x305edc8cdd04718b17fa2d35cd00051e7780bb71b93058c8727f6dd04bb9aa57
  Account ID:
0x305edc8cdd04718b17fa2d35cd00051e7780bb71b93058c8727f6dd04bb9aa57
  SS58 Address: 5DA8LQVJwNssXkrvMpsSCmZXUTMay2iznqkJsdJth5A7LkWM

...

Step 3: Now please copy the ss58 Address for the corresponding key from the above-generated data and paste it in the place provided below for each key.

Validator 1:

Grandpa: 5FNFwatBDgwnGPjuZ8X4hqjzai2khjH1auorsKgBtjprBu53
Stash: 5EhBdmSa8xU8kehBWf1bY8FAxxbqGKUEsZ2X1visyqRMxYE4
Controller: 5EZq6KtccJuf9CsHDSM1ecmP9kCRhnuxjp46oWLEW14B9YwL
Babe: 5G1aN9JCMpPtZte4XNFseiu1fC9XnkmbHBgvtxTw8DQhcpbH
imonline: 5ELhP5dCJBZzkKS9Ee1f9bxfrnpWdM3qC4uWVFLSQTZjm1Nx
authdiscovery: 5DDZLoCoMN6LPc2hsmTpiHQNUkD7R1YGR9Soee8HqADbrzwJ

Validator 2:

Grandpa: 5DA3rYQPr1Ecsn9fJANwGE5ijQ495mPwD5YpoTKVPKYpqQi4
Stash: 5Chk2Xb3EnyADkDb42uW63aMdoyqPw1cDzJM4w4aeio264rk
StashEth: 0xeB581B4E5e81Af1543b2438b2f9541A015316866
Controller: 5Dz4225UnPnVaUKmojKhJkWfnr2NWFPvEpgPH7sqZUaoMQgg
ControllerEth: 0x99f31feba83EE0232D6dB76E7643612D533513b9
Babe: 5GEKrLEqmRLiPQDqpXkDNgKjBTrwSHqP9nRL6gweYrSukaAs
imonline: 5FWq4ZY5AQxXfigZxS4r4nR8xwaf22NiVVqz43gEtbHRTJbz
authdiscovery: 5DSPnW4YSJ6USGftfTgPWwAsY1bWXn7Z2LAXLyNfx9AkWtVZ

Validator 3:

Grandpa: 5GBbLxtC3jZsnrxn7pojbWYaVHJF2oK8hmEqKwERgJV9kuDD
Stash: 5D9hVaxRH8eR9vRtXnw9iwzQ5ebvdKyw9dJz4r41nHGCd85g
StashEth: 0x99dF3aE935D5A83954D8635Bf9f2BAE95500b50a
Controller: 5H1c9Ypc1ShkaXrsdGBrMPi7LZPiEmF4cBzKeEaiELo5KMKY

ControllerEth: 0x7Df97005a105756f5058208e599271b19861B0C1
Babe: 5GYqXxNrUZmheY1Qt9K1wKhvYS6afCrmMwyxiNeAabGRnwg8
imonline: 5DoGdqPrrUzdAF379Jp4vPhJc1ZtZ9yuUF8ujRoFnq9Ct1hc
authdiscovery: 5H1dXgYwaq8MMVhKmyziN1tSsZhV16iqKzu92uFKHzkupiaa

Step 2:

(ii) Create the Substrate environment on the local computer (Follow the steps here).
(iii) Build Substrate Binary .
(iv) Create the Node keys (Following below steps

Step 3:
Step-1: Generate 3 Node keys using the following command -
subkey generate-node-key

By running this command you will get a key pair like this -
12D3KooWPYwihLEGLXiwi2JjcWXxyvN6QcX5Gahhz8mZFN65TUHy
1428aa848e75f46734d681fae845cab1b9ae0aac1d91e39f561f136d197dbbae

The first key is the public node key and the second one is the private node key.
Step-2: Please keep the key pair for all 3 validators in a secure place.
Step-3: Please provide the public node keys for each validator in the following block -
Node Key Validator 1 - 12D3KooWDgXPdVefQ5JMuoV6eVrCmwRd3PCWrEyvLQrr6BDLz3kk
Node Key Validator 2 - 12D3KooWCEAykLooj1dvviXGoq5fn7A5Aeg5PSXciP3ctWusrWkA
Node Key Validator 3 - 12D3KooWMfep2X4vs5XmMGbS4T69mLdmhQmuXWnmX5yfNxkKu5Rj

Step 4:
Build the `ChainSpec.json` file using the following command -
./target/release/clarus-node build-spec --chain clarus > clarusChainSpec.json

./target/release/clarus-node build-spec --raw --chain clarusChainSpec.json > clarusChainSpecRaw.json

Step 5:
Update keys in ChainSpecs file

Step 6: Run node

Node 1:

nohup ./clarus-node --chain /home/ubuntu/ChainSpecRaw.json --base-path /home/ubuntu/POS1/ --name clarus-pos-node1 --in-peers 256 --validator --rpc-cors all --rpc-methods=safe --unsafe-rpc-external --unsafe-ws-external --port 30335 --ws-port 9945 --rpc-port 9933 --pruning=archive --no-telemetry --detailed-log-output --allow-private-ipv4 >> /home/ubuntu/node1.log 2>&1 &

./insert-nodekey.sh "$CHAIN" "/home/ubuntu/POS1" "$V1_GRAN" "$V1_BABE" "$V1_IMOL" "$V1_AUDI"

Node 2:
nohup ./clarus-node --chain /home/ubuntu/ChainSpecRaw.json --base-path /home/ubuntu/ --name clarus-pos-node2 --in-peers 256 --validator --rpc-cors all --rpc-methods=safe --unsafe-rpc-external --unsafe-ws-external --port 30336 --ws-port 9946 --rpc-port 9936 --offchain-worker Always --pruning=archive --no-telemetry --detailed-log-output --allow-private-ipv4 --reserved-nodes /ip4/3.65.42.47/tcp/30335/p2p/12D3KooWDcZgqxfMFTpBDAQBD4RCr1m9FHeKmtzDi1h8BhdhkWRG >> /home/ubuntu/node2.log 2>&1 &

./insert-nodekey.sh "$CHAIN" "/home/ubuntu/POS2" "$V2_GRAN" "$V2_BABE" "$V2_IMOL" "$V2_AUDI"

Node 3:
nohup ./clarus-node --chain /home/ubuntu/ChainSpecRaw.json --base-path /home/ubuntu/POS3/ --name clarus-pos-node3 --in-peers 256 --validator --rpc-cors all --rpc-methods=safe --unsafe-rpc-external --unsafe-ws-external --port 30337 --ws-port 9947 --rpc-port 9937 --pruning=archive --no-telemetry --detailed-log-output --allow-private-ipv4 --reserved-nodes /ip4/3.65.42.47/tcp/30335/p2p/12D3KooWDcZgqxfMFTpBDAQBD4RCr1m9FHeKmtzDi1h8BhdhkWRG >> /home/ubuntu/node3.log 2>&1 &

./insert-nodekey.sh "$CHAIN" "/home/ubuntu/POS3" "$V3_GRAN" "$V3_BABE" "$V3_IMOL" "$V3_AUDI"

==========

Run a validator

To run a validator node you need an account with at least `100,000` MYID. Now follow the below steps to run a validator node.

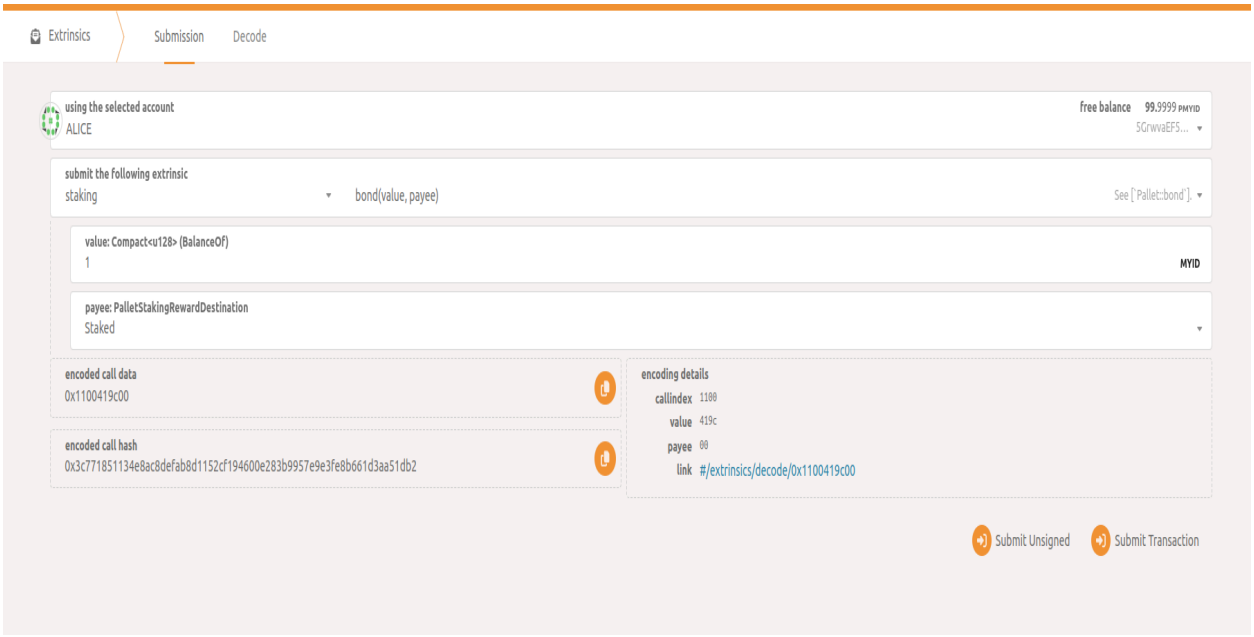
1. Put both `clarus-node` binary and the `ChainSpecRaw.json` file in the same directory.
2. Run the below command to start a node.

```
clarus-node \
  --chain ./ChainSpecRaw.json \
  --base-path ./data \
  --name awesome-node \ # Change this to any name you want
  --in-peers 256 \
  --validator \
  --rpc-cors all \
  --rpc-methods=safe \
  --port 30335 \
  --rpc-port 9945 \
  --offchain-worker Always \
  --pruning=archive  \
  --no-telemetry \
  --detailed-log-output \
  --allow-private-ipv4 \
  --reserved-nodes path # this is optional if path already mentioned in chainSpecs
```

3. Go to this link in your browser.

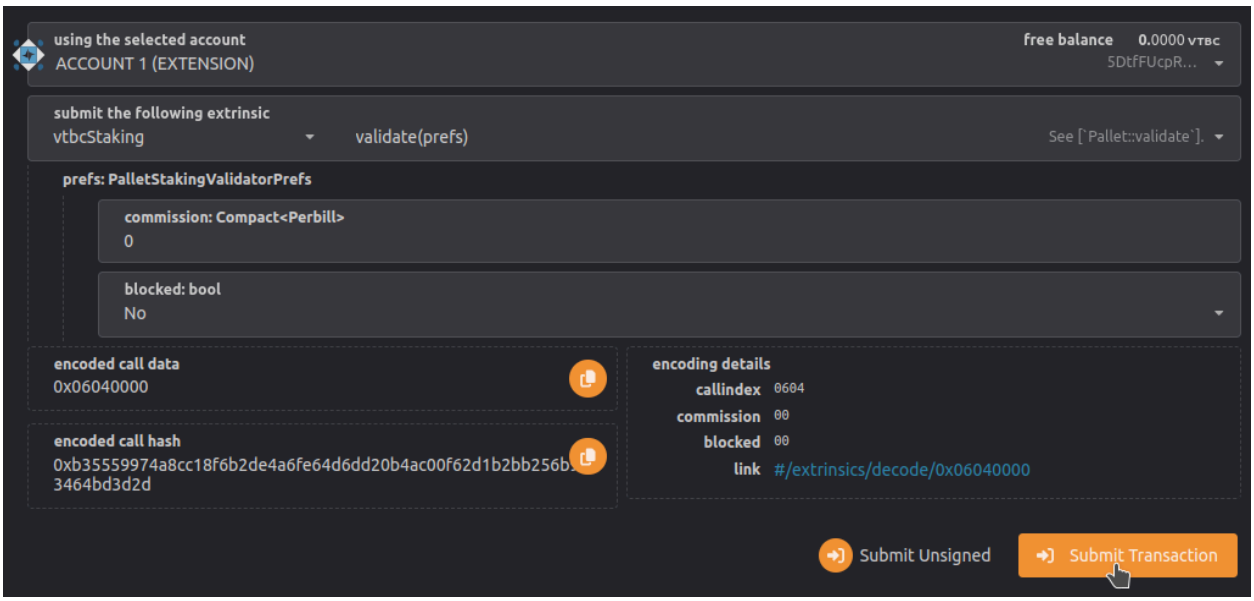
4. Go to `Developers → Extrinsics` section.

    1. Select your account using `the selected account` field. In `submit the following extrinsic` field, select `Staking` from the left menu, and select `bond(value, payee)` from the right menu. In the `value` field, enter `100000` , which is equal to 100,000 MYID. In the `payee` field, select `Stash` . After that, the UI should look like this.

- 

2. Now click the `Submit Transaction` button, and then click `Sign and Submit` . It should open a popup asking for your account password. Enter the password and click `Sign the transaction` .

3. Now select `validate(prefs)` instead of `bond(value, payee)` .You can keep the default values in `commission` and `blocked` fields. Submit and sign the transaction again.

- 

4. Now open a terminal and run the following command.

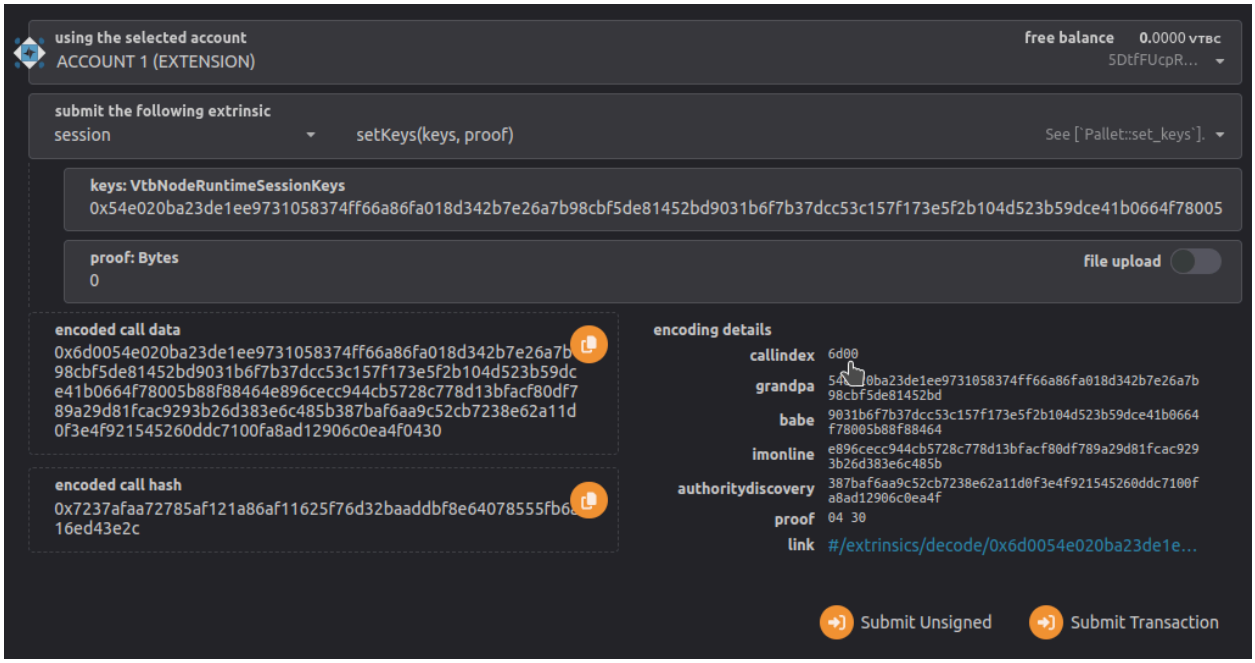
   curl -H "Content-Type: application/json" -d '{"id":1, "jsonrpc":"2.0", "method": "author_rotateKeys", "params": []}' http://localhost:9945/

   This will print a response similar to this.

   {"jsonrpc":"2.0","result":"0x54e020ba23de1ee9731058374ff66a86fa018d342b7e26a7b98cbf5de81452bd9031b6f7b37dcc53c157f173e5f2b104d523b59dce41b0664f78005b88f88464e896cecc944cb5728c778d13bfacf80df789a29d81fcac9293b26d383e6c485b387baf6aa9c52cb7238e62a11d0f3e4f921545260ddc7100fa8ad12906c0ea4f","id":1}

   Copy the value of the `result` field, In this example a value starting with `0x54e` and ending with `ea4f` . Your result will show a different value.

5. Now go back to the browser. Go to `Developer → Extensions` section, and this time select `session` instead of `Staking` , and on the right side select `setKeys(keys, proof)` , In the `keys` field past the value you copied earlier, and in the `proof` field enter `0`. It should look something like this.

- 

   Now submit and sign the transaction again.

   And that is all, from now on you may be selected as a validator from the next era.