Beginner's Guide - How to Setup a Quilibrium CeremonyClient node - Dawn
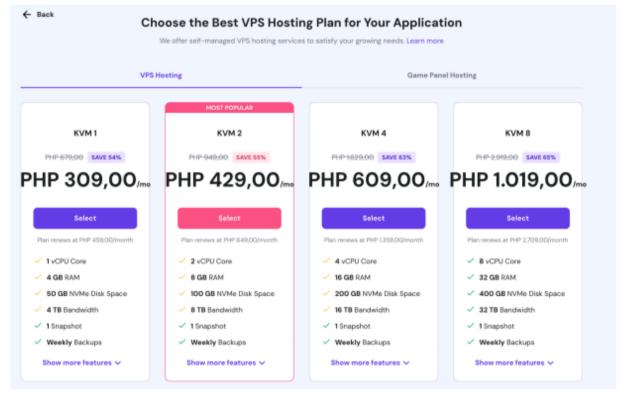
Created by @demipoet

Let's connect at Farcaster ->

1. Secure your hardware

    a. Get a VPS running Ubuntu 22.04
        i. My recommendations
            1. Hostinger.com (My referral link)
            2. I recommend getting the KVM8 for Dawn
            3. In USD - KVM 8 costs $18.30/mo

← Back

Choose the Best VPS Hosting Plan for Your Application

We offer self-managed VPS hosting services to satisfy your growing needs. Learn more

VPS Hosting | Game Panel Hosting

	MOST POPULAR		
KVM 1	**KVM 2**	**KVM 4**	**KVM 8**
PHP 679,00 SAVE 54%	PHP 949,00 SAVE 55%	PHP 1,629,00 SAVE 63%	PHP 2,919,00 SAVE 65%
PHP 309,00/mo	**PHP 429,00/mo**	**PHP 609,00/mo**	**PHP 1.019,00/mo**
Select	Select	Select	Select
Plan renews at PHP 459,00/month	Plan renews at PHP 649,00/month	Plan renews at PHP 1.359,00/month	Plan renews at PHP 2.709,00/month
✓ 1 vCPU Core	✓ 2 vCPU Core	✓ 4 vCPU Core	✓ 8 vCPU Core
✓ 4 GB RAM	✓ 8 GB RAM	✓ 16 GB RAM	✓ 32 GB RAM
✓ 50 GB NVMe Disk Space	✓ 100 GB NVMe Disk Space	✓ 200 GB NVMe Disk Space	✓ 400 GB NVMe Disk Space
✓ 4 TB Bandwidth	✓ 8 TB Bandwidth	✓ 16 TB Bandwidth	✓ 32 TB Bandwidth
✓ 1 Snapshot	✓ 1 Snapshot	✓ 1 Snapshot	✓ 1 Snapshot
✓ Weekly Backups	✓ Weekly Backups	✓ Weekly Backups	✓ Weekly Backups
Show more features ⌄	Show more features ⌄	Show more features ⌄	Show more features ⌄

    b. After buying, go to https://hpanel.hostinger.com/servers/plans
    c. Select **KVM 8** only. KVM 2 will not work for the Quilibrium Dawn testnet looks to be especially different from other networks so this requires KVM 8 😉 It requires if VPS at least 8vCPU Cores and 16GB RAM

> 8 vCPU Core
> 32 GB RAM
> 400 GB NVMe Disk Space
> 32 TB Bandwidth
> 1 Snapshot
> Weekly Backups

    d. You can select a plan duration (monthly/yearly) after selecting KVM 8
        i. Then link your card to pay or pay with crypto
        ii. Note that the discounted rate may only be available on the yearly plan
        iii. Then follow installation instructions

e. Connect to your VPS
  i. Downloading and install PuTTY: http://putty.org/ or just use your Terminal if you are running MacOS or Linux
  ii. Put your IP address into terminal client & click open Click "Accept" Type "root" & click enter Type your password & click enter
  iii. SSH into your VPS

2. How to SSH for the first time to your new VPS
  a. ssh root@<ip_address_of_VPS>
  b. yes

```
                    -MBP ~ % ssh root@19
The authenticity of host '19             (              )' can't be established.
ED25519 key fingerprint is SHA256:nQY                                    jo.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

  c. After typing yes, press enter and you should see the message below.

```
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '193.          ' (ED25519) to the list of known hosts.
Connection closed by 193.         port 22
```

  d. Type ssh root@<ip_address_of_VPS> again
  e. Enter your root password

3. Prerequisite software

  a. Type sudo apt -q update
  b. Install git
  i. sudo apt install git -y
  ii. git --version
  iii. It must show
      git version 2.34.1

  c. Install golang 1.20.14. The release right now (as of 1.2.7) does not support its project builds with 1.21.x
  i. wget https://go.dev/dl/go1.20.14.linux-amd64.tar.gz or https://go.dev/dl/go1.20.14.linux-arm64.tar.gz (the project supports both AMD64 and ARM64)
  ii. sudo tar -xvf go1.20.14.linux-amd64.tar.gz
  iii. sudo mv go /usr/local
  iv. sudo rm go1.20.14.linux-amd64.tar.gz
  v. sudo vi ~/.bashrc

      vi.   Press i to start inserting text inside the file

     vii.   At the end of the file, add these lines

          1. GOROOT=/usr/local/go

          2. GOPATH=$HOME/go

          3. PATH=$GOPATH/bin:$GOROOT/bin:$PATH

    viii.   Press esc to stop the insert mode

     ix.   :wq

      x.   On command line, run source ~/.bashrc

     xi.   go version

     xii.   It must show

               go version go1.20.14 linux/amd64

4. Configure Linux network device settings

   a. Purpose: to optimize throughput (bandwidth) and latency for large parallel jobs typical of networks like Q

   b. Run:

      i.   vim /etc/sysctl.conf

     ii.   Go to the end of the file

    iii.   press i to start inserting text

    iv.   Copy and paste the 3 lines below into the file. The values below are six hundred million.

```
# Increase buffer sizes for better network performance
net.core.rmem_max=600000000
net.core.wmem_max=600000000
```

     v.   Press esc

     vi.   Type :wq to save and exit

    vii.   sudo sysctl -p

   c. Reboot your VPS, by running reboot

5. Clone the Quilibrium Ceremony Client repo from github

   a. SSH into your VPS again (ssh root@<ip_address_of_VPS>)

   b. git clone https://github.com/QuilibriumNetwork/ceremonyclient.git

   c. cd ceremonyclient/node

6. **Start the node** [DEVELOPMENT]

    a. Go to ceremonyclient/node folder

    b. If you have a voucher from the offline ceremony, first run:

        i. Run: sudo vim /root/voucher.hex

        ii. Press i to start inserting text

        iii. Copy-paste the 228-character voucher hex into the file

        iv. Press esc

        v. Type :wq! and press enter

        vi. GOEXPERIMENT=arenas go run ./... -import-priv-key `cat /root/voucher.hex`

        vii. Please take note that the above character is ` and not a '

        viii. I found out that that makes a difference

    c. If you do not have a voucher, or have already run the above, run:

        i. GOEXPERIMENT=arenas go run ./...

    d. You should see this as proof you have started your ceremonyclient node successfully

7. Get your Node Peer ID

    a. Go to ceremonyclient/node folder

    b. Run:

        i. GOEXPERIMENT=arenas go run ./... -peer-id

    c. You should see the response below

```
[root@vmi1628339:~/ceremonyclient/node# GOEXPERIMENT=arenas go run ./... -peer-id]":"ceremony/
Peer ID:                                                                          X7JfvdvQg+hk
root@vmi1628339:~/ceremonyclient/node#
```

    d. Save your peer ID for reference

8. Enable gRPC for your Ceremony Client node (To enable gRPC calls possible)

    a. *Please Note:* This interface, while read-only, is unauthenticated and not rate-limited. It is recommended that you only enable them if you are properly controlling access via firewall or only query via localhost.

    b. Stop the node first by pressing Ctrl + C on your keyboard

    c. Go to ceremonyclient/node folder

    d. Run:

        i. sudo vi .config/config.yml

        ii. Press i to start editing config.yml file

        iii. On the line listenGrpcMultiaddr: "", replace it with

           listenGrpcMultiaddr: /ip4/127.0.0.1/tcp/8337

        Side note: Yes. It is really without " " and instead enters the value /ip4/127.0.0.1/tcp/8337 straight away. We have verified that this is working and correct, as the response below, once you have done Step 17, is not possible to show if it is not.

```
[root@vmi1688661:~# grpcurl -plaintext localhost:8337 quilibrium.node.node.pb.NodeService.GetNodeInfo
{
  "peerId": "Qm                                       5",
  "maxFrame": "2754"
}
root@vmi1688661:~#
```

     iv.   [Optional] Enabling RESTMultiplier is optional, but if you want to enable it, replace listenRESTMultiaddr: "" with listenRESTMultiaddr: /ip4/127.0.0.1/tcp/8338

```
22|   maxFrames: -1
B0:   pendingCommitWorkers: 4
ed    mi̲nimumPeersRequired: 0
m     statsMultiaddr: ""
b     difficulty: 0
st  db:
]     path: .config/store
    listenGrpcMultiaddr: /ip4/127.0.0.1/tcp/8337
    listenRESTMultiaddr: ""
    logFile: ""
```

     v.   Press :wq to save and exit

e. Note: Ensure that port 8337 among other needed ports are enabled via firewall. Since VPS is on Ubuntu, enable ufw. Run the following commands:

     i.   sudo ufw enable

     ii.   sudo ufw allow 22

     iii.   sudo ufw allow 8336 - default port used by Q node

     iv.   sudo ufw allow 8337

     v.   sudo ufw allow 8338 - only if you did step 7.c.iv.

     vi.   sudo ufw status should show you below response

```
[root@vmi1628339:~/ceremonyclient/node# vi .config/config.yml
[root@vmi1628339:~/ceremonyclient/node# sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
22                         ALLOW       Anywhere
8336                       ALLOW       Anywhere
8337                       ALLOW       Anywhere
8317                       ALLOW       Anywhere
8316                       ALLOW       Anywhere
22 (v6)                    ALLOW       Anywhere (v6)
8336 (v6)                  ALLOW       Anywhere (v6)
8337 (v6)                  ALLOW       Anywhere (v6)
8317 (v6)                  ALLOW       Anywhere (v6)
8316 (v6)                  ALLOW       Anywhere (v6)

root@vmi1628339:~/ceremonyclient/node#
```

f. Restart your node by running:

     i.   GOEXPERIMENT=arenas go run ./...

g. Let the node run for 10-30 minutes before being able to check your QUIL balance below

9. Get your $QUIL balance [DEVELOPMENT] - [deprecated]

a. While the Development node is running with log trace, open another shell/terminal window and ssh again into your VPS
b. On the 2nd window, go to ceremonyclient/node folder
c. Run:
    i.   GOEXPERIMENT=arenas go run ./... -balance
d. You should see the response below

DEPRECATED

```
root@vmi1628339:~/ceremonyclient/node# GOEXPERIMENT=arenas go run ./... -balance er":"ceremo
Confirmed balance: 0 QUIL                                                          UybnbBt6X7J
root@vmi1628339:~/ceremonyclient/node#
```

a. **-balance tag**: The -balance command tag when calling the GOEXPERIMENT=arenas go run ./... **-balance has been deprecated** since Dawn v1.2.9. Unless your node was upgraded to the latest version, yet upgraded from a version <1.2.9 (e.g. 1.2.7) (i.e. the very first version your node ran with was one older than 1.2.9), the -balance command tag will not work anymore and will face the error below.

```
root@vmi1688531:~/ceremonyclient/node# GOEXPERIMENT=arenas /root/go/bin/node -balance
panic: error getting token info: rpc error: code = Unknown desc = get token info: get highest candidate data clock frame: item not found

goroutine 1 [running]:
main.main()
        /root/ceremonyclient/node/main.go:101 +0x7f4
root@vmi1688531:~/ceremonyclient/node#
```

b. **GetTokenInfo gRPCurl function**: The same error shows for the GetTokenInfo function if you were to call using gRpcUrl library (see image below)

```
        /root/ceremonyclient/node/main.go:101 +0x7f4
root@vmi1688531:~/ceremonyclient/node# grpcurl -plaintext localhost:8337 quilibrium.node.node.pb.NodeService.GetTokenInfo
ERROR:
  Code: Unknown
  Message: get token info: get highest candidate data clock frame: item not found
root@vmi1688531:~/ceremonyclient/node#
```

c. **Solution**: You can still see your balance in the logs of a running node. In later releases, it is certain there will be a friendlier way to see a node's $QUIL balance.

10. Safekeep and Backup your Private Keypair (IMPORTANT!!)

a. Backup the 2 files (i.e. either copy the 2 files to another folder inside root or open them with vim editor and copy paste their text in a text file in your laptop (separate machine from your VPS))

i. ~/ceremonyclient/node/.config/config.yml
    ii. ~/ceremonyclient/node/.config/keys.yml
  b. The keys.yml contains your private keypair
  c. The key in the config.yml encrypts the keypair in node/.config/keys.yml
     (future key management options 🔜)

11. Stop the node [DEVELOPMENT]

  a. In the window where this screen is running,

  b. Stop the node first by pressing Ctrl + C on your keyboard

12. Build the node binary file in /root/go/bin folder
  a. GOEXPERIMENT=arenas go install ./...
  b. This will build and compile the app binary in /root/go/bin/ folder
  c. If you are not using the root user, the binary will be found in
     /home/<user>/go/bin folder

13. Start the Quilibrium Node as a Service [PRODUCTION]

  a. **Prerequisite** - Make sure you perform the Step 11 before going here, as
     the go run command will build and compile the node app binary in
     /root/go/bin/ folder (if you did not use 'root' user, but another custom
     superuser profile, the node app binary can be found in
     /home/<user>/go/bin/ folder)
  b. Go to /root folder (if you used another profile, go to /home/<user> folder
  c. Create the service configuration file

9

       i.   Run:

            1. vi /lib/systemd/system/ceremonyclient.service to create the file
            2. Press i to start inserting text
            3. Enter the following text inside the ceremonyclient.service file

```
[Unit]
Description=Ceremony Client Go App Service

[Service]
Type=simple
Restart=always
RestartSec=5s
WorkingDirectory=/root/ceremonyclient/node
Environment=GOEXPERIMENT=arenas
ExecStart=/root/go/bin/node ./...

[Install]
WantedBy=multi-user.target
```

               Note: if you did not use the root user, but another custom superuser profile, change the WorkingDirectory=/home/<user>/ceremonyclient/node and ExecStart=/home/<user>/go/bin/node ./...

            4. Press the esc button
            5. Type :wq and press enter to save and exit the file

   d. Start the node service

       i.   Run:

            1. service ceremonyclient start

   e. Do not do this now, but just fyi - You can **stop the node service** any time

       i.   Run:

            1. service ceremonyclient stop

14. View the trace/info logs of your CeremonyClient service [PRODUCTION]

   a. Make sure your ceremonyclient service is activated
   b. Run:
       i.   service ceremonyclient status
   c. It should show the following

```
[root@vmi1628339:~/ceremonyclient/node# service ceremonyclient status
● ceremonyclient.service - Ceremony Client Go App Service
     Loaded: loaded (/lib/systemd/system/ceremonyclient.service; disabled; vendor preset: enabled)
     Active: active (running) since Sat 2024-02-24 01:56:08 CET; 3h 45min ago
   Main PID: 7062 (node)
      Tasks: 11 (limit: 7037)
     Memory: 2.2G
        CPU: 13h 15min 29.836s
     CGroup: /system.slice/ceremonyclient.service
             └─7062 /root/go/bin/node ./...

Feb 24 05:41:53 vmi1628339.contaboserver.net node[7062]: {"level":"info","ts":1708749713.99216,"caller":"ceremony/consen>
Feb 24 05:41:53 vmi1628339.contaboserver.net node[7062]: {"level":"info","ts":1708749713.9924552,"caller":"ceremony/cons>
Feb 24 05:41:53 vmi1628339.contaboserver.net node[7062]: {"level":"info","ts":1708749713.994163,"caller":"ceremony/conse>
Feb 24 05:41:54 vmi1628339.contaboserver.net node[7062]: {"level":"info","ts":1708749714.120332,"caller":"ceremony/conse>
Feb 24 05:41:54 vmi1628339.contaboserver.net node[7062]: {"level":"info","ts":1708749714.1210737,"caller":"ceremony/cons>
Feb 24 05:41:54 vmi1628339.contaboserver.net node[7062]: {"level":"info","ts":1708749714.121471,"caller":"ceremony/cerem>
Feb 24 05:41:54 vmi1628339.contaboserver.net node[7062]: {"level":"info","ts":1708749714.806582,"caller":"ceremony/cerem>
Feb 24 05:41:54 vmi1628339.contaboserver.net node[7062]: {"level":"info","ts":1708749714.807488,"caller":"ceremony/cerem>
Feb 24 05:41:54 vmi1628339.contaboserver.net node[7062]: {"level":"info","ts":1708749714.8076587,"caller":"ceremony/cere>
Feb 24 05:41:54 vmi1628339.contaboserver.net node[7062]: {"level":"info","ts":1708749714.8106244,"caller":"ceremony/cere>
lines 1-20/20 (END)
^C
root@vmi1628339:~/ceremonyclient/node#
```

d. Press ctrl+c to exit the status log
e. Run this command to start looking at your node [info] logs
    i. sudo journalctl -u ceremonyclient.service -f --no-hostname -o cat
f. It should show the following. I usually maintain 2 terminal windows, one for logs, one for running commands.

```
{"level":"info","ts":1708750054.3016915,"caller":"ceremony/consensus_frames.go:154","msg":"checking peer list","peers":8768,"uncooperative_peers":0,"current_head_frame":275}
{"level":"info","ts":1708750054.302748,"caller":"ceremony/consensus_frames.go:187","msg":"polling peer for new frames","peer_id":"EiCzRAa31a+y7G30e8arvcyVUT4q1tCD0pIh2rXukIUBpA=="}
{"level":"info","ts":1708750054.4039736,"caller":"ceremony/consensus_frames.go:238","msg":"received compressed sync frame","from":0,"to":0,"frames":0,"proofs":0}
{"level":"info","ts":1708750054.4041598,"caller":"ceremony/consensus_frames.go:154","msg":"checking peer list","peers":8768,"uncooperative_peers":0,"current_head_frame":275}
{"level":"info","ts":1708750054.4052343,"caller":"ceremony/consensus_frames.go:187","msg":"polling peer for new frames","peer_id":"EiCzRAa31a+y7G30e8arvcyVUT4q1tCD0pIh2rXukIUBpA=="}
{"level":"info","ts":1708750054.5158002,"caller":"ceremony/consensus_frames.go:238","msg":"received compressed sync frame","from":0,"to":0,"frames":0,"proofs":0}
{"level":"info","ts":1708750054.5159855,"caller":"ceremony/consensus_frames.go:154","msg":"checking peer list","peers":8768,"uncooperative_peers":0,"current_head_frame":275}
{"level":"info","ts":1708750054.517625,"caller":"ceremony/consensus_frames.go:187","msg":"polling peer for new frames","peer_id":"EiCzRAa31a+y7G30e8arvcyVUT4q1tCD0pIh2rXukIUBpA=="}
{"level":"info","ts":1708750054.6421385,"caller":"ceremony/consensus_frames.go:238","msg":"received compressed sync frame","from":0,"to":0,"frames":0,"proofs":0}
{"level":"info","ts":1708750054.6427758,"caller":"ceremony/consensus_frames.go:154","msg":"checking peer list","peers":8768,"uncooperative_peers":0,"current_head_frame":275}
{"level":"info","ts":1708750054.6446133,"caller":"ceremony/consensus_frames.go:187","msg":"polling peer for new frames","peer_id":"EiCzRAa31a+y7G30e8arvcyVUT4q1tCD0pIh2rXukIUBpA=="}
{"level":"info","ts":1708750054.7417095,"caller":"ceremony/consensus_frames.go:238","msg":"received compressed sync frame","from":0,"to":0,"frames":0,"proofs":0}
{"level":"info","ts":1708750054.7422693,"caller":"ceremony/consensus_frames.go:154","msg":"checking peer list","peers":8768,"uncooperative_peers":0,"current_head_frame":275}
{"level":"info","ts":1708750054.7441568,"caller":"ceremony/consensus_frames.go:187","msg":"polling peer for new frames","peer_id":"EiCzRAa31a+y7G30e8arvcyVUT4q1tCD0pIh2rXukIUBpA=="}
{"level":"info","ts":1708750054.8031204,"caller":"ceremony/consensus_frames.go:238","msg":"received compressed sync frame","from":0,"to":0,"frames":0,"proofs":0}
{"level":"info","ts":1708750054.8033295,"caller":"ceremony/consensus_frames.go:154","msg":"checking peer list","peers":8768,"uncooperative_peers":0,"current_head_frame":275}
{"level":"info","ts":1708750054.8050833,"caller":"ceremony/consensus_frames.go:187","msg":"polling peer for new frames","peer_id":"EiCzRAa31a+y7G30e8arvcyVUT4q1tCD0pIh2rXukIUBpA=="}
{"level":"info","ts":1708750054.8739676,"caller":"ceremony/consensus_frames.go:238","msg":"received compressed sync frame","from":0,"to":0,"frames":0,"proofs":0}
{"level":"info","ts":1708750054.8741555,"caller":"ceremony/consensus_frames.go:348","msg":"returning leader frame","frame_number":275}
{"level":"info","ts":1708750054.8741987,"caller":"ceremony/ceremony_execution_engine.go:574","msg":"evaluating next frame","frame_number":275}
{"level":"info","ts":1708750055.5361543,"caller":"p2p/blossomsub.go:528","msg":"connected to peer","peer_id":"QmQy3BQbLCt4MrDxU4UAGnWXnH2VzgtN2jENd7RFDtF24q"}
{"level":"info","ts":1708750055.53705,"caller":"p2p/blossomsub.go:519","msg":"found peer","peer_id":"Qne4vVWH2tKssPsK6DfQMvWHYNbAdHckiTerP8KxkHSxQ4"}
{"level":"info","ts":1708750055.6913543,"caller":"ceremony/ceremony_execution_engine.go:592","msg":"current application state","my_balance":0,"lobby_state":"OPEN"}
{"level":"info","ts":1708750055.6914527,"caller":"ceremony/ceremony_execution_engine.go:613","msg":"lobby open for joins","joined_participants":0,"preferred_participants":41,"in_lobby":false,"state_count":0}
{"level":"info","ts":1708750055.6914737,"caller":"ceremony/ceremony_execution_engine.go:625","msg":"joining lobby","proving_key":"AaSQSGp0T8Y7/iZOJI1AFKNbLNjfFq+OKH+Lt/vJIG4ky38F5lPKzahPymIzf54rPcog+RIcs5UA"}
{"level":"info","ts":1708750055.6938298,"caller":"ceremony/ceremony_execution_engine.go:636","msg":"preparing contribution"}
```

g. Press ctrl+c to exit the node [info] log **OR** you can maintain this window and leave it open, and just open and SSH into your VPS using a 2nd window wherein you can execute further commands (e.g. -balance)

15. Get your $QUIL balance [PRODUCTION]

  a. Go to ceremonyclient/node folder
  b. Run:
    i. GOEXPERIMENT=arenas /root/go/bin/node -balance
    ii. or GOEXPERIMENT=arenas /home/<user>/go/bin/node -balance if you used another user profile instead of root
  c. You should see the response below

```
root@vmi1628339:~/ceremonyclient/node# GOEXPERIMENT=arenas /root/go/bin/node -balance
Owned balance: 0 QUIL
Unconfirmed balance: 0 QUIL
root@vmi1628339:~/ceremonyclient/node#
```

16. Upgrading your Q node to latest release

  a. Run: service ceremonyclient status
  b. If the response says service is Active, run: service ceremonyclient stop first. If the response says service is Inactive, proceed.
  c. Go to ceremonyclient folder
  d. Check whether your local branch of the ceremonyclient repo is up to date.
    i. Run:
      1. git fetch origin
    ii. If the result certain files were downloaded or updated, then you need to do a git merge command
  e. Download the latest version
    i. Run:
      1. git merge origin
  f. Do a go clean to clear all previous build files
    i. Go to ceremonyclient/node folder
    ii. Run:
      1. GOEXPERIMENT=arenas go clean -v -n -a ./...
  g. Remove the compiled go binary file
    i. Run:
      1. rm /root/go/bin/node
      2. Or rm /home/<user>/go/bin/node if you used another user profile

h. Do step 11 (*Build the node binary file in /root/go/bin folder*) above again
  i. Step 11 will build and compile the /root/go/bin/node binary file
i. Make sure the build file is indeed there
  i. Run:
    1. ls /root/go/bin
    2. or ls /home/<user>/go/bin if you used another user profile instead of root
  ii. It should show you

```
root@vmi1628339:~/ceremonyclient/node# ls /root/go/bin/
node
root@vmi1628339:~/ceremonyclient/node#
```

j. Start the node service
  i. Run:
    1. service ceremonyclient start
  ii. Do Step 13 (*View the trace/info logs of your CeremonyClient service*) again to ensure your service running correct

17. Using gRPCurl For More Information on Q Network

a. Instruction: This section is not part of the installation or setup process for the Q Node. The commands within this section are used only after you have successfully completed the steps above and you just want to query for more information about the node you are running and the $QUIL related questions.

b. Reference : https://github.com/mscurtescu/ceremonyclient/wiki/gRPCurl-How-To

c. Installation

  i. There are many ways to install gRPCurl (brew, docker, Go) and since we already have installed Go you may install gRPCurl from the source package
    1. Run:
    go install github.com/fullstorydev/grpcurl/cmd/grpcurl@latest
  ii. The command above takes care of installing the command in the bin sub-folder of $GOPATH
  iii. Verify your installation works.
    1. Run:
    grpcurl -help

d. Available gRPC Functions to call in Quilibrium Network

i. Get Node Info
1. Run:

grpcurl -plaintext localhost:8337 quilibrium.node.node.pb.NodeService.GetNodeInfo

2. Output:

```
{
  "peerId": "FiBgOwjQo6aam6iQAqlfvc6wJ3xnTXNTzOLDLwFHyBuujw==",
  "maxFrame": "7658"
}
```

ii. Get Token Info
1. Run:

grpcurl -plaintext localhost:8337 quilibrium.node.node.pb.NodeService.GetTokenInfo

2. Output:

```
{
  "confirmedTokenSupply": "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAACsuy4PqzEAA=",
  "unconfirmedTokenSupply": "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAACsuxtRfqwAA=",
  "ownedTokens": "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA=",
  "unconfirmedOwnedTokens": "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA="
}
```

DEPRECATED

1. **See Step 9 for more details**

iii. Count of all Q Nodes
1. Run:

grpcurl -plaintext -max-msg-sz 5000000 localhost:8337
quilibrium.node.node.pb.NodeService.GetPeerInfo | grep peerId | wc -l

2. Output:

```
10000
```

iv. Get Peer Info
1. Run:

grpcurl -plaintext -max-msg-sz 5000000 localhost:8337
quilibrium.node.node.pb.NodeService.GetPeerInfo | less

2. Output (truncated):

```
{
  "peerInfo": [
    {
      "peerId": "EiCIlJTLaQ+GE0ElmoCTblwK/OWfWTzDA9hClAyAmNNCew==",
```

```
      "multiaddrs": [
        ""
      ],
      "maxFrame": "1439",
      "timestamp": "1708984499614",
      "version": "AQIP",
      "signature":
"uOXC/WFc3tCsaqJKplle2q/oeUDGlwZf/ZrDdjWqn/Z3ZBUZZy2jwbB78vhFrNsArVuEw+x9eSWAVT
q7ugE4ME5bUNUB4+7Gl6BtV8vPNwqh5NWbfNooKbHDNyees6MEi+VsM04wS4wf147SLiGPxicA",
      "publicKey":
"gMTEXXTwVMd9JGBTdJ3+eh/qIGuYapU+PGe7kg3HUgdlj4Ff8NBMVhHxoN/dYLnmpaquGOcqgKcA",
      "totalDistance":
"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAET2HAgKu7BSR42UPY8
a2MEk/qOTrCjR9shges9bZvO/Ng=="
    },
    {
      "peerId": "EiA3XnxlesdF2BJXwEuVFXw+gj5g5ss4ajJBQ3TNXWiAxA==",
      "multiaddrs": [
        ""
      ],
      "maxFrame": "6783",
      "timestamp": "1704564573411",
      "version": "AQIB",
      "signature":
"JOJoCRUmzizAlFjf3dXB8wAeEvFXsXCI166A/sLu4hjIJktYzvZPAc83abLeG3Zm8WHhuhtnHH4AXH
itiWGlqwtFbWCRE6TuDFioemJBKhRgwS3bLF3KIC0yWEcBSTM5hgJCDWe7oCSI2NV8n7mufRUA",
      "publicKey":
"prlfQeX4gLPLOz87HChdlzOWihxXeoVdUBMdBTjNmSYsYAGnJuMmGDdgZoipZWsrgsEbhnjKfa2A",
      "totalDistance":
"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAHW0BTyd89JROjwc23I
0r7qOJ7ROny+L+EWDln6OHytKKw=="
    },
```

18. Frequently Asked Questions (FAQ)

   a. What's my balance? Why am I still at 0?

      i. This question is the most frequently asked of them all. Depending on when you joined in on the project, the answer will vary. At this point, fully-synchronized nodes should start accumulating a balance. We are working to improve synchronization so it's not an incredibly long slog to get there. Everyone who has been participating should see some balance when Dusk launches.

   b. How do I know my node is operating properly?

      i. You should see the frame number your node is processing continually increase. This may be slow at first due to heavy fork reconciliation – think of each node with a prover key as being able to send valid frames, but the timereel as a means to choose between them. Because we have had many iterations and lots of early networking issues, we may be initial forks that your node must churn through. We're working on improving sync to be smarter about this so you don't have to reconcile all of them.

   c. Where do I run a node?

      i. Don't run it on IaaS/PaaS providers like AWS/GCP/Azure. The egress fees will eat you alive. If you don't want to/can't run on your own hardware, VPS/ bare metal hosts do well. Be advised: Q uses a lot of bandwidth in this stage.

   d. How do I run a node?

      i. If you have git and go 1.20 installed, it's just these three steps:

         1. git clone https://github.com/QuilibriumNetwork/ceremonyclient.git

         2. cd ceremonyclient/node

         3. GOEXPERIMENT=arenas go run ./...

      ii. If you don't, follow whatever guide for your operating system to install go 1.20 (MUST be 1.20.X, 1.19- won't work, 1.21+ won't either).

   e. I'm getting cannot load arena: malformed module path "arena" when I run it, what am I doing wrong?

      i. Wrong version of golang. Make sure you installed 1.20 (and it has priority on your PATH environment variable)

   f. I'm getting the error below when I run the -balance command flag, what am I doing wrong?

```
panic: error getting token info: rpc error: code = Unknown desc = get
token info : get highest candidate data clock frame: item not found


goroutine 1 [running]:
main.main()
        /root/ceremonyclient/node/main.go:101 +0x745
exit status 2
```

     i.   Nothing. See Step 9 for more details.

g. Someone is offering me QUIL OTC, is this legit?
     i.   No. Tokens do not unlock until Dusk, nobody can transfer you QUIL or buy it from you.

h. Who is developing Quilibrium?
     i.   Officially the core Quil dev team is just Cassie as of right now (making this an oddly third person statement), plus many folks who have submitted PRs to the core project, as well as related projects like Agost with quilibrium-rs, and Sir0uk with the nodekeeper dashboard. If you'd like to join in, there's loads of things to work on, please DM!

i. Wen balance
     i.   See question a.