Setting up a Verifier Node on MacOS

Questions about this guide? Please contact Oneiro Support.

The first step in running your own node is getting a dedicated computer. An ndau node doesn't need a lot of computational power, but it does needs to be running 24/7. In our experience setting up personally run nodes, we like Mac Minis since they're energy efficient, easy to set up and reliable.

Minimum Specs: OS: macOS Catalina, Processor: 1.4 GHz Dual Core Intel Core i5, Memory: 4 GB

For reference see this doc on GitHub:
https://github.com/ndau/commands/blob/ejm-documentation/docker/node_operator.md

1) Make sure MacOS is updated to latest version.

2) Install GitHub Desktop
https://desktop.github.com/

3) Install Docker Desktop
Follow instructions here: https://docs.docker.com/docker-for-mac/install/
After installing, launch Docker Desktop and complete the Docker Quick Start walkthrough.
Demo Container success screen should look like this:

4) Install Homebrew
Use these instructions: https://docs.brew.sh/Installation

5) Install jq
In MacOS terminal window:

```
ndau-node@ndau-nodes-Mac-mini / % brew install jq
```

6) Clone the ndau/commands repo

```
ndau-node@ndau-nodes-Mac-mini / % git clone https://github.com/ndau/commands
```

7) Launch the docker container
Open MacOS Terminal.

Go to the commands directory you just cloned:

```
ndau-node@ndau-nodes-Mac-mini bin % cd commands/docker/bin
```

Set the following environment variables at the command line, setting $NODENAME to the name you'd like to use for your node and $NDAU_NETWORK to the name of the network you'd like to connect to

```
Give your node a name.
NODENAME=my-node

Choose the network - mainnet, testnet, devnet, localnet
NDAU_NETWORK=mainnet

Choose the ports you would like to use for...
...communication with other nodes on the network.
...responding to RPC requests to your node.
...and responding to ndau API requests to your node.
P2P_PORT=26665
RPC_PORT=26675
API_PORT=3035
```

Choose a name for your node and the network (testnet or mainnet) you'd like to connect to. Run the container from the command line and watch it spin up for the first time. The Docker image tag (here `ndauimage:fa2c4ad` will change as the ndau node software is updated. Your node will start using the most recent block-numbered snapshot of the network, so that value (here `snapshot-mainnet-82622`) will change).

```
ndau-node@ndau-nodes-Mac-mini bin % ./runcontainer.sh $NDAU_NETWORK $NODENAME
$P2P_PORT $RPC_PORT $API_PORT
Network: mainnet
Container: my-node
P2P port: 26665
RPC port: 26675
API port: 3035
Snapshot: (latest)
Fetching https://s3.us-east-2.amazonaws.com/ndau-json/services.json...
Testing connection to peer mainnet-0.ndau.tech:26661...
Connection to mainnet-0.ndau.tech port 26661 [tcp/*] succeeded!
Getting peer info for https://mainnet-0.ndau.tech:26670...
Peer id: 49b08ed396e190717e6c2c64620cb4529f3519a7
Testing connection to peer mainnet-1.ndau.tech:26661...
Connection to mainnet-1.ndau.tech port 26661 [tcp/*] succeeded!
```

```
Getting peer info for https://mainnet-1.ndau.tech:26670...
Peer id: 3823adecffd2d8bd7cd4ff9752ff70de5b889af8
Testing connection to peer mainnet-2.ndau.tech:26661...
Connection to mainnet-2.ndau.tech port 26661 [tcp/*] succeeded!
Getting peer info for https://mainnet-2.ndau.tech:26670...
Peer id: d6333397905f8d35504007f0aaf235039a2fbc2d
Testing connection to peer mainnet-3.ndau.tech:26661...
Connection to mainnet-3.ndau.tech port 26661 [tcp/*] succeeded!
Getting peer info for https://mainnet-3.ndau.tech:26670...
Peer id: 10127e995c8dba09fe8352d345f9add0d7e8297a
Testing connection to peer mainnet-4.ndau.tech:26661...
Connection to mainnet-4.ndau.tech port 26661 [tcp/*] succeeded!
Getting peer info for https://mainnet-4.ndau.tech:26670...
Peer id: 9ccc286a6099b2187b349d38db55f9d81ac6c5ea
Persistent peers:
'49b08ed396e190717e6c2c64620cb4529f3519a7@mainnet-0.ndau.tech:26661,3823adecffd2d
8bd7cd4ff9752ff70de5b889af8@mainnet-1.ndau.tech:26661,d6333397905f8d35504007f0aaf
235039a2fbc2d@mainnet-2.ndau.tech:26661,10127e995c8dba09fe8352d345f9add0d7e8297a@
mainnet-3.ndau.tech:26661,9ccc286a6099b2187b349d38db55f9d81ac6c5ea@mainnet-4.ndau
.tech:26661'
Stopping my-node...
done
Fetching current-mainnet.txt...
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100     8  100     8    0     0     41      0 --:--:-- --:--:-- --:--:--    41
Unable to find ndauimage:fa2c4ad locally; fetching...
Fetching ndauimage-fa2c4ad.docker.gz...
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  136M  100  136M    0     0  16.4M      0  0:00:08  0:00:08 --:--:-- 19.5M
Loading ndauimage:fa2c4ad...
f1b5933fe4b5: Loading layer [==================================================>]
5.796MB/5.796MB
e461f61726d7: Loading layer [==================================================>]
70.81MB/70.81MB
e8ad0876ecf2: Loading layer [==================================================>]
52.72MB/52.72MB
98d40738e079: Loading layer [==================================================>]
40.45kB/40.45kB
c77f14c0e32a: Loading layer [==================================================>]
28.07MB/28.07MB
d5c8f6d5c179: Loading layer [==================================================>]
22.1MB/22.1MB
b4b40d9116c4: Loading layer [==================================================>]
195.3MB/195.3MB
Loaded image: ndauimage:fa2c4ad
Creating container...
d068e9088deb95b509441d551328e43b9c62dde53a10f26a95814f291ac4d562
Starting container...
```

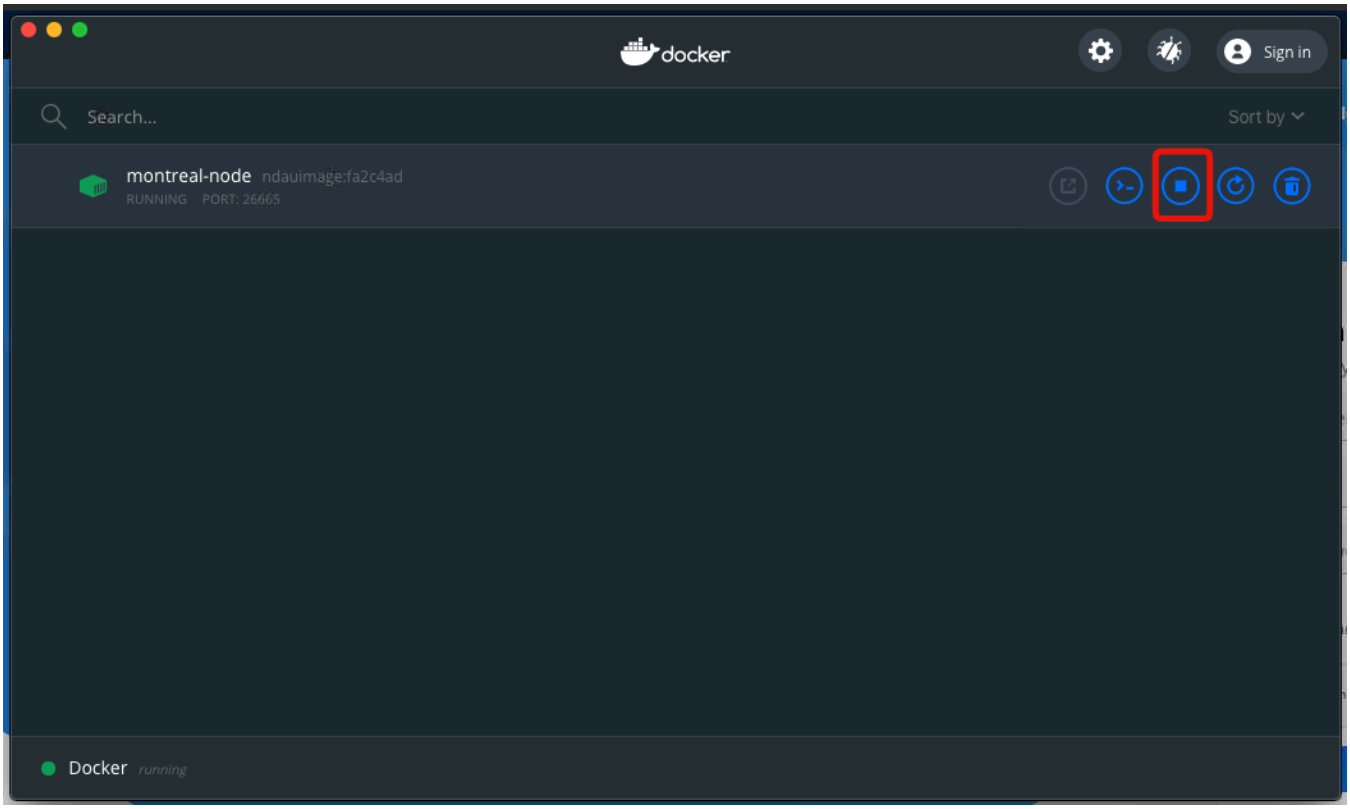
```
montreal-node
Waiting for the node to fully spin up...
Watching montreal-node...
Node is ready; dumping container logs...
> montreal-node is starting up
> Configuring node group...
> Fetching latest-mainnet.txt...
> Fetching snapshot-mainnet-82622.tgz...
> Extracting snapshot-mainnet-82622.tgz...
> Validating /image/snapshot...
> No node identity found; a new node identity will be generated
> Using ndau config file: /image/docker-config-mainnet.toml
> Webhook config pre-copy:
>   NodeRewardWebhookDelay = 10.0
> Webhook config post-copy:
>   NodeRewardWebhook =
"https://7ovwffck3i.execute-api.us-east-1.amazonaws.com/mainnet/claim_winner"
>   NodeRewardWebhookDelay = 10
> Webhook config post-sed:
>   NodeRewardWebhook =
"https://7ovwffck3i.execute-api.us-east-1.amazonaws.com/mainnet/claim_winner"
>   NodeRewardWebhookDelay = 10.0
> Configuring tendermint...
> I[2020-01-30|15:29:34.305] Generated private validator
module=main keyFile=/image/data/tendermint/config/priv_validator_key.json
stateFile=/image/data/tendermint/data/priv_validator_state.json
> I[2020-01-30|15:29:34.305] Generated node key
module=main path=/image/data/tendermint/config/node_key.json
> I[2020-01-30|15:29:34.305] Found genesis file
module=main path=/image/data/tendermint/config/genesis.json
> Configuration complete
> Converting persistent peer domain names to IP addresses...
> Using IP 52.0.58.109 for peer
49b08ed396e190717e6c2c64620cb4529f3519a7@mainnet-0.ndau.tech:26661
> Using IP 3.15.50.243 for peer
3823adecffd2d8bd7cd4ff9752ff70de5b889af8@mainnet-1.ndau.tech:26661
> Using IP 52.8.209.126 for peer
d6333397905f8d35504007f0aaf235039a2fbc2d@mainnet-2.ndau.tech:26661
> Using IP 54.191.172.56 for peer
10127e995c8dba09fe8352d345f9add0d7e8297a@mainnet-3.ndau.tech:26661
> Using IP 52.74.90.120 for peer
9ccc286a6099b2187b349d38db55f9d81ac6c5ea@mainnet-4.ndau.tech:26661
> logging to /image/logs
> Started procmon as PID 246
> Waiting for node group...
> Waiting for valid block height...
> Generating identity file...
> Node group montreal-node is now running

The node identity has been generated and copied out of the container here:
```

```
/Users/ndau-node/Documents/GitHub/commands/docker/bin/node-identity-montreal-node
.tgz

You can always get it at a later time by running the following:
  docker cp montreal-node:/image/node-identity.tgz node-identity.tgz
It can be used to restart this container with the same identity it has now
Keep it secret; keep it safe

done
ndau-node@ndau-nodes-Mac-mini bin %
```

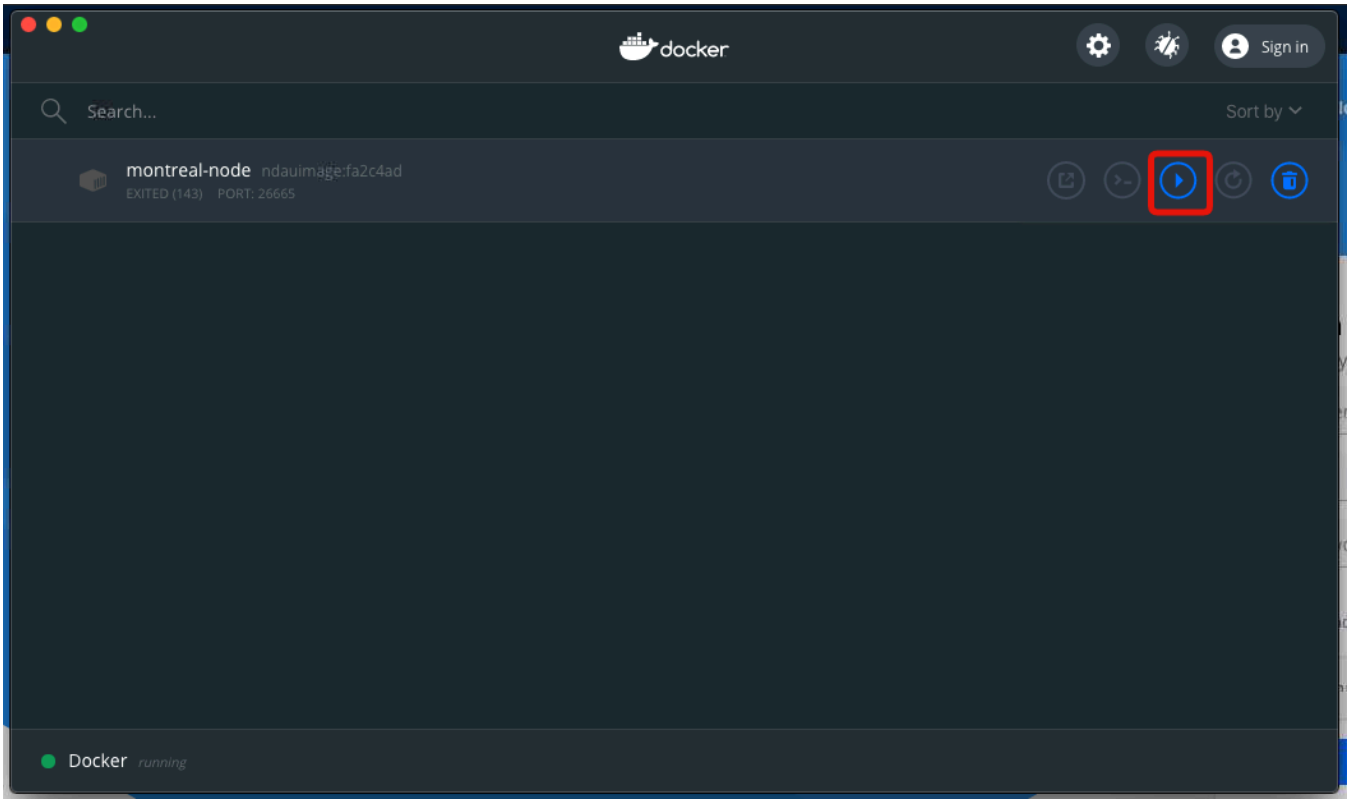
8) Check the status of your node here:
http://localhost:3035/node/status

9) CONGRATULATIONS! YOU ARE DONE!
You can Stop and Start your node from now on using the Docker Desktop:

Stopping your node:



Starting your node:


docker
Sign in
Search...
Sort by
montreal-node ndauimage:fa2c4ad
EXITED (143) PORT: 26665
Docker running