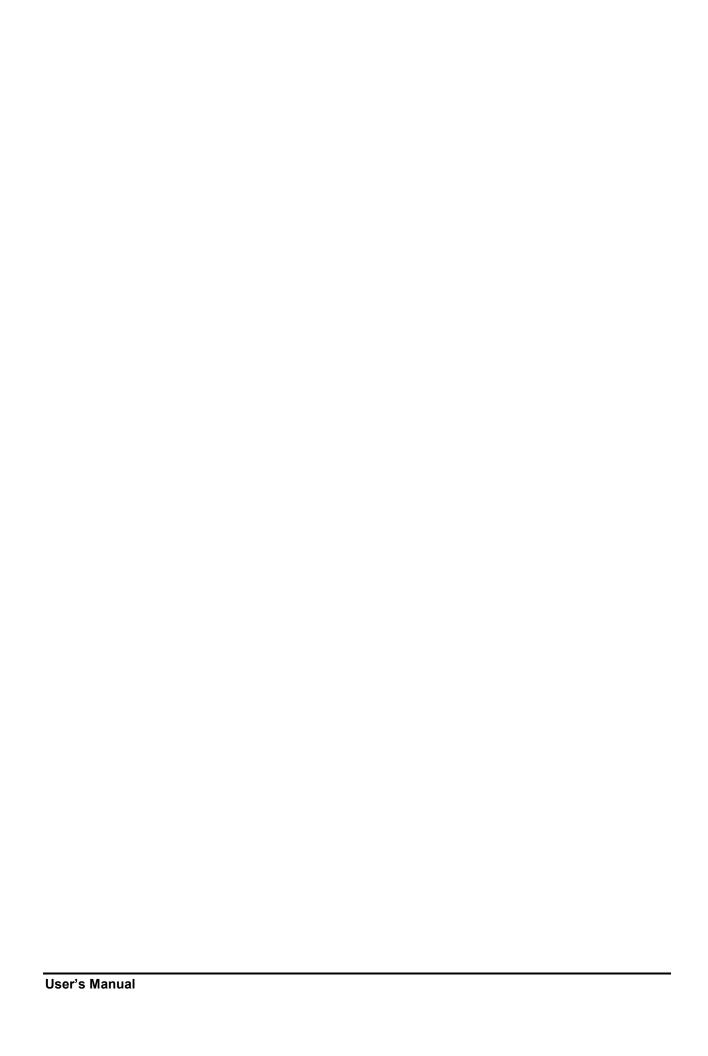# Aion Network

Owner's Manual

The Aion Foundation

April 2018

Aion Owner's Manual
Table of Contents

A. General Information

1.1 Aion Overview

The Aion multi-tier blockchain network is like a computer network, providing a protocol and standard for dissimilar systems to communicate. However, in addition to information, the Aion network will pass logic and value among participating blockchains to create a contiguous value chain where every transaction occurs on-chain, with logic and value passing among chains as freely as liquid assets.

Aion is a multi-tier blockchain network that will enable any private or public-sector organization to:
- Federate: Send data and value between any Aion-compliant blockchain and Ethereum.
- Scale: Provide fast transaction processing and increased data capacity to all Aion blockchains.
- Spoke: Allow the creation of customized public or private blockchains that maintain interoperability with other blockchains, but allow publishers to choose governance, consensus mechanisms, issuance, and participation.

1.2 AION Kilimanjaro Release

The Aion Kilimanjaro blockchain is the genesis implementation of the interoperability blockchain. It is designed to be a fair, distributed, open blockchain architecture that is capable of fulfilling the requirements specified in the multi-tier blockchain network architecture. As an open blockchain, Aion Kilimanjaro was designed with the following core components:
- Virtual Machine – FastVM with EVM Source Compatibility
- Aion Proof of Work – Equihash210_9
- Aion Core – Multi-chain Framework, Wire Protocol (P2P), Tx Pool, Event Manager
- Aion APIs – Java API, Web3 API Compatibility
- Aion Interchain – Token Bridge and Interchain Communication

1.3 Using the Owner's Manual

This owner's manual is intended as a walkthrough guide to help users install Aion and start using its various features. It is intended for non-developers who want to rapidly start participating in the Aion Network ecosystem. This can be in the form of running an Aion node, mining, creating a local wallet, transacting Aion and more. Once you are ready, you can go to Getting Started to download the binary. You can also visit the tutorials within the docs for specific examples on the

user functionalities. For further information, please read through the FAQs section to find common issues, concerns and general questions. We want the information in this manual to be accessible in a single place, streamlining the experience of first-time users of Aion.

Please visit the Error Handling section in the Appendix to reach the Aion team for any additional questions/concerns.

1.4   Links

For more information on Aion, the architecture, and a deeper dive into the components, please check out the Aion Technical Introduction here.

To access Aion's source code and begin contributing to this open-source project, please visit the Aion Github Wiki here.

B.   Getting Started

2.1   System requirements

Below are the requirements for setting up and running the kernel.

- Ubuntu 16.04 or a later version
- CUDA 9.0 compatible driver (NVIDIA 387.34+) for GPU based mining. **(optional)**

2.2   Installation

2.2.1 KERNEL

Getting the kernel

You can acquire access to the most recent Aion binary file here.

Once you have the binary file, you will need to unarchive the contents in your chosen destination using one of the following methods:

A. Right click on the file and select *Extract Here.*
B. Open a terminal and run the following command:

   tar xvjf aion-{version}.tar.bz2

A folder named *aion* will be created in the chosen directory.

Configuration

- **Creating accounts *(only keystore file is supported)***

Account addresses are required for mining and sending transactions. An address is a hashed representation of the users publicKey, prefixed with an identifier "0xA0".. You can create accounts on the kernel by following these steps:

1. Go to *aion* directory
2. Open a terminal and run the following command:

   ./aion.sh -a create

3. Input your chosen password once prompted
4. Copy the created account address shown.

After account creation, an associated keystore file is generated and placed in the *aion/keystore* directory. Each keystore file name contains the address and can be referred to in case of lost account address.

Note
You should backup the keystore folder if you want to keep the generated keys.

- **Generating peer Id (optional)**

The default config will auto assign an ID on the first boot, and will continue to use that ID afterwards.

If you want to generate a custom ID, you can manually enter a valid UUID into the config file. NodeID can be changed any point.

- **Setting up seed nodes (optional)**

Your kernel will have access to the seed nodes by default. You should not remove these nodes from the configuration. Additional peers can be added to configuration by updating the *nodes* section and including peer id, port, and IP. Peer Id for a node should be generated using the steps mentioned above.

```
<nodes>
   <node>p2p://PEER_ID@IP:PORT</node>
</nodes>
```

- **Setting up IP**

To allow connection among peers the *ip* section of the config file should be updated to include a public IP. If you are unsure about having a public IP, set it to 0.0.0.0.

```
<p2p>
   <ip>0.0.0.0</ip>
   <port>30303</port>
</p2p>
```

Note
UPnP is disabled by default; therefore, you will be able to connect to the network without being discoverable. If you want to be discoverable to the network, activate the port forwarding and enable UPnP to allow nodes to receive inbound connections, which can also help with network connectivity.

Launch kernel

Go to the *aion* directory and run the following command in the terminal to launch the kernel.

./aion.sh

Note
You can set show-status to true in your config file to prompt the network to display connected peers.

<show-status>true</show-status>

Alternatively, you can open another terminal and run the command below to view connected peers:

netstat -antp | grep java

2.2.2 MINER

There are two types of miners that can work with the kernel, Internal and External miner.

Internal miner

The internal miner runs on the local node and is provided with the kernel. It can be configured by modifying the *config.xml* file in aion directory (aion/config/config.xml). In order to setup internal mining, the following fields should be updated in *consensus* section:
- Mining: should be set to true to enable mining
- Miner-address: the wallet address that will collect AION for mining blocks. The account address created in creating accounts section can be used for this purpose.
- Cpu-mine-threads: number of logical CPU cores to use for mining. This number should be between 1 and 75% of your maximum CPU logical cores. The number of logical cores may be seen in either task manager (Windows) or system monitor (Ubuntu). It is not recommended to go above 75% of the total logical cores.
- extraData: any hex string up to 32 bytes

A sample config file following internal mining configuration is shown below. Once the config file has been updated, the kernel should pick up the mining settings next time it is launched. Mining is normally delayed 10 seconds to allow sufficient time for the kernel to fully start.

<consensus>
  <mining>true</mining>
<miner-address>0000000000000000000000000000000000000000000000000000000000000111</miner-address>
  <cpu-mine-threads>2</cpu-mine-threads>
  <extra-data>MyAion</extra-data>
</consensus>

External miner

CPU and GPU-CUDA miners are provided as options for external mining. These miners are designed to be used in conjunction with the provided reference mining pool. Future miner versions will be adapted to join public pools. The provided reference pool requires more advanced Ubuntu configuration; if you are not comfortable with the Ubuntu terminal we recommend using the internal kernel miner.

Follow these steps to run the external miner:
1. Download a pre-built miner binary from the aion_miner release page.
2. Once you have the binary file, open a terminal and run the following command:

   tar xvjf aionminer-{type}-{version}.bz2
3. Run the *aionminer* file by setting up desired parameter values. Sample commands to run the miner are as follows:

   CPU

   Following the instructions runs AION CPU miner with 4 threads connecting to a mining pool running locally, listening on port 3333 for incoming connections.

   ./aionminer -t 4 -l 127.0.0.1:3333 -u {0xacc}

   Here is an example of running a benchmark on your CPU using a single thread:
   ./aionminer -b -t 1

   GPU

   Following the instructions runs AION CUDA miner with 64 blocks and 64 threads per block on device 0 using solver version 1 (CUDA Tromp).

   ./aionminer -cd 0 -cv 1 -cb 64 -ct 64 -u {0xacc}

   Here is an example of running a benchmark on your GPU (GPU 0, CUDA-Tromp solver):

   ./aionminer -cd 0 -cv 1 -cb 64 -ct 64 -b

   General miner parameters:

   -h [help]   *Print this help message and quits*
   -l [location]   *Stratum server:port*
   -u [username]   *Username (aion address)*
   -a [port]   *Local API port (default: 0 = do not bind)*

| *-d [level]* | *Debug print level (0 : print all, 5 : fatal only, default: 2)* |
| *-b [--benchmark] [=arg(=200)]* | *Run in benchmark mode (default: 200 iterations)* |

Extra parameters for CPU miner:

| *-t [threads]* | *Number of CPU threads* |
| *-e [ext]* | *Force CPU ext (0 = SSE2, 1 = AVX, 2 = AVX2)* |

Extra parameters for CUDA miner:

--ci	*Show CUDA info*
--cv	*CUDA solver (0 = djeZo, 1 = tromp, default=1)*
--cd	*Enable mining on spec. devices*
--cb	*Number of blocks (per device)*
--ct	*Number of threads per block (per device)*

2.2.3 MINING POOL

The Aion mining pool should be used in conjunction with the Aion mining client on the Aion network. During the initial Kilimanjaro launch, only the Aion solo miner is currently officially supported on the network. Users can join public pools at their own risk. The Aion Foundation may release an official mining pool at a later date.

The primary difference in the solo pool is that it cannot distribute the reward. All rewards will be added to the account specified in the Kernel configuration file.

Prerequisites

- Aion kernel
- Python v2.7
  - Included by default with Ubuntu desktop, may need to be installed separately in Ubuntu server. Run the following commands to install:

    sudo apt-get update
    sudo apt-get install build-essential

- make
  - Included by default with Ubuntu desktop, may need to be installed separately in Ubuntu server. Run the following command to install:

    sudo apt-get install python2.7 python-dev

Follow these steps to run the mining pool:

1. Open the Aion configuration file located in *aion/config/config.xml*

2. Update the *consensus* section:
    a. Mining: should be set to false to disable internal mining
    b. Miner-address: the address which will receive mined block rewards. The address is a 64 character (32 byte) hex string containing the public key and address of an account. This account will start with a0.

    E.g.
    <consensus>
        <mining>false</mining>
        <miner-address>A012341234123412341234123412341234123412341234123412341234123 4123412</miner-address>
            <cpu-mine-threads>8</cpu-mine-threads>
            <extra-data>AION</extra-data>
    </consensus>

3. Ensure that the APIs enabled section includes the "stratum" option

    E.g.
    <apis-enabled>web3,eth,personal,stratum</apis-enabled>

4. Download the latest prepackaged aion_solo_pool from its release page.
5. Navigate to the download directory, open a terminal and run the following command to unpack the file:

    tar xf aion-solo-pool-<VERSION>.tar.gz

6. Run the configure script to download and build all of the pool dependencies and place them into the current directory. This script may take several minutes to complete, however it must only be run once. Configure script can be run using:
    ./configure.sh

7. Run the solo_pool using the quickstart run script. This script will start and stop both the pool and redis server.

    ./run_quickstart.sh

8. Open another terminal in *aion* directory and start the kernel:

    ./aion.sh

9. The pool is now ready to accept incoming client connections and to distribute work to clients.

C.   Using Aion

After successfully setting up your Aion node, you can start using Aion using the Web3 console. The Web3 console facilitates the usage of Web3 application programming interface on top of Aion network.

Prerequisites

Following are the prerequisites for building Aion Web3 API. You can download and install each item by following the provided link.

- node.js version 8.9.1 ([download and install](#))
- npm version 5.5.1 (Typically included with node installation.)
- gulp version 3.9.1 ([download and install](#))

Installation

Web3 console can be downloaded by visiting its [repository](#) on GitHub. You can download a zip file containing the repository by pressing the "*Clone or download"* button and extracting the zip file contents into your desired directory. Alternatively, the repository can be cloned through the terminal using:

git clone https://github.com/aionnetwork/aion_web3

Navigate to the *aion_web3* directory and run the following commands:

npm install

gulp build

Or

npm install --save aion-web3

3.1   API Use

You can include Aion's web3 API in your application using one of these methods:

- Web use
   include *dist/web3.min.js* in your html file

- Node use:
   var Web3 = require('aion-web3')
   var web3 = new Web3(new Web3.providers.HttpProvider("http://localhost:8545"));

3.2   Running Web3 console

Navigate to the *aion* folder and run the kernel using:

./aion.sh

Navigate to the *aion_web3* folder and and run the console in another terminal:

node console.js

You can now interact with Aion kernel using the console. Below are some sample operations that can be performed using the console.

3.2.1 VIEW LOCAL ACCOUNTS

You can view the accounts that are created and stored locally using the following command.

personal.listAccounts

Any of the accounts shown can be used to perform mentioned commands below. Refer to the configuration section of kernel for more information regarding creating accounts.

3.2.2 CHECK YOUR BALANCE

In order to view the balance in an account, run the following command and replace the *0xacc* with account address:

eth.getBalance('{0xacc}')

E.g.
*eth.getBalance('0xa01234123412341234123412341234123412341234123412341234123
412')*

3.2.3 UNLOCK YOUR ACCOUNT

In order to make a transaction, the **sender** should first unlock their account. In order to unlock an account, run the following command and replace the *0xacc, pwd, secondsToUnlock* with account address, associated password and unlock duration in milliseconds:

personal.unlockAccount('{0xacc}', '{pw}', {secondsToUnlock})

E.g.
*personal.unlockAccount('0xa01234123412341234123412341234123412341234123412341234
1234123412, 'abc', 60000)*

If unlocking is successful, the output will be *true*; otherwise *false*.

Note
The user should only attempt this command with their <u>local</u> node that they manage/own themselves. _

3.2.4 TRANSFER AION

You can send a transaction from one account to another with a specified value. It is important to note that the account of sender should be <u>unlocked</u> before making the transfer. Replace *{0xacc_from}*, *{0xacc_to}*, *{amount_to_transfer}* with the sender's unlocked account address, recipient's account address, and amount to transfer.

eth.sendTransaction({from: '{0xacc_from}', to: '{0xacc_to}', value: {amount_to_transfer}})

E.g.
eth.sendTransaction({from:'
0xa01234123412341234123412341234123412341234123412341234123412', to:
'0xabcdabcdabcdabcdabcdabcdabcdabcdabcdabcdabcdabcdabcdabcdabcd'
, value: 10000)

3.2.5 CREATE WALLET BACKUP OR IMPORT ACCOUNT

We recommend you keep track of your keystore files and copy them over to the new binary to successfully import/export accounts.

D.   Appendix

4.1   Error Handling

To report any issues after running and using the Aion Kernel, please find the following resources:

- Github (https://github.com/aionnetwork): you can log tickets and report bugs/issues on the AionNetwork Github page.
- Forum (https://forum.aion.network/): check out our forum to chat with the Aion community and moderators to stay up to date with latest updates, issues, resolutions and discussions.
- Email us at support@aion.network for any specific questions

4.2   FAQs

<u>General</u>

- **Where do I start?**
Go to our <u>releases</u> page to download the latest binary and then go to the <u>Getting Started</u> section to set up your Aion node.

- **How can I help?**
You can directly contribute to Aion during our testnet phase by running a node during our testnet phase. We encourage all users to actively participate in the various error handling channels we have available to help us fix any issues.

Additionally, there will be many other areas community members can contribute to the project in the near future; from content translations, posting feature reviews/requests, reinforcing community engagement, to 'retweets', writing blog posts, and industry analysis.

- **How do I find the latest version and how often does it get updated?**
Each week, the latest version of the binary will be sent out via the support email to the signed-up users.

- **Can I run this on a virtual machine, raspberry pi, vagrant box?**

Yes, but they are not supported. If you follow the instructions of setting it up with a standard VM, it should work without problems. The Aion Kenel is untested on Raspberry Pi and will not be supported for mining (could work for syncing and light mining).

- **Where do I report errors?**

Refer to the previous section of error handling for details.

- **Is there a bounty/reward program?**

Yes, there will be a reward program to incentivize early miners and users of Aion to actively participate in the early stages of the Kilimanjaro launch. More details on the bounty structure will be released shortly.

Installation

- **How do I install Aion and Web3 interface?**

Please refer to the "Installation" section under "Setting up your Aion node" for the detailed steps.

- **How do I monitor the Testnet installation?**

No installation, it is a compiled binary therefore it is just executed.

- **What are the system requirements to install and run the Kernel?**

You will need Ubuntu 16.04 or a later version. Please refer to the "Setting up your Aion node" section for more details.

Mining

- **How do I know that I'm mining?**

You will see a CPU output with "x" number of solutions produced per second.

- **How many CPU threads should I allocate for my mining node?**

You should allocate between 1 to a maximum of 75% of capacity for running a node. Anything above the maximum is not advised.

- **How many GPU threads/blocks should I allocate for my mining node?**

These values will largely depend on your GPU; benchmarking on a 1050 TI has found a good balance will be reached with 64 blocks and 64 threads per block. We recommend running the CUDA miner in benchmark mode with a variety of parameters to find the best set of parameters for your GPU.

Synchronization

- **How do I know I am syncing?**

You should see the following output:

18-02-22 11:33:33.148 INFO SYNC [sync-import]: <import-best num=818 hash=ef5b2b txs=0>
18-02-22 11:33:33.390 INFO SYNC [sync-import]: <import-best num=819 hash=09b0bc txs=0>
18-02-22 11:33:33.533 INFO SYNC [sync-import]: <import-best num=820 hash=58f004 txs=0>

- **How much longer do I have to sync for?**

In the config file you can change log of SYNC to Debug and can find out the best block of the node you are syncing with.

- **I am getting an error that I'm not synced, what should I do to get synced again?**

Restart the node *(Ctrl+C to stop, ./aion.sh to start)*, if the error persists, try to revert to previous blocks (click here for details). If your issue has still not been resolved, you may need to delete the database folder if the database is corrupted beyond the Aion Kernel's repair capabilities.