# Creating a Private Network using Ethash (Proof of Work) Consensus Protocol

A private network provides a configurable network for testing. By configuring a low difficulty and enabling mining, blocks are created quickly.

You can test multi-block and multi-user scenarios on a private network before moving to one of the public testnets.

**Important**

An Ethereum private network created as described here is isolated but not protected or secure. We recommend running the private network behind a properly configured firewall.

## Prerequisites

[Pantheon](https://docs.pantheon.pegasys.tech/en/stable/Installation/Install-Binaries/)

[Curl (or similar web service client)](https://curl.haxx.se/download.html)

## Steps

To create a private network:

1. [Create Folders](https://docs.pantheon.pegasys.tech/en/stable/Tutorials/Create-Private-Network/#1-create-folders)
2. [Create Genesis File](https://docs.pantheon.pegasys.tech/en/stable/Tutorials/Create-Private-Network/#2-create-genesis-file)
3. [Get Public Key of First Node](https://docs.pantheon.pegasys.tech/en/stable/Tutorials/Create-Private-Network/#3-get-public-key-of-first-node)
4. [Start First Node as Bootnode](https://docs.pantheon.pegasys.tech/en/stable/Tutorials/Create-Private-Network/#4-restart-first-node-as-bootnode)
5. [Start Node-2](https://docs.pantheon.pegasys.tech/en/stable/Tutorials/Create-Private-Network/#5-start-node-2)
6. [Start Node-3](https://docs.pantheon.pegasys.tech/en/stable/Tutorials/Create-Private-Network/#6-start-node-3)
7. [Confirm Private Network is Working](https://docs.pantheon.pegasys.tech/en/stable/Tutorials/Create-Private-Network/#7-confirm-private-network-is-working)

### 1. Create Folders

Each node requires a data directory for the blockchain data. When the node is started, the node key is saved in this directory.

Create directories for your private network, each of the three nodes, and a data directory for each node:

Private-Network/  
├── Node-1  
│ ├── Node-1-data-path  
├── Node-2  
│ ├── Node-2-data-path  
└── Node-3  
 ├── Node-3-data-path

### 2. Create Genesis File

The genesis file defines the genesis block of the blockchain (that is, the start of the blockchain). The genesis file includes entries for configuring the blockchain such as the mining difficulty and initial accounts and balances.

All nodes in a network must use the same genesis file. The [network ID](https://docs.pantheon.pegasys.tech/en/stable/Configuring-Pantheon/NetworkID-And-ChainID/) defaults to the chainID in the genesis file. The fixeddifficulty enables blocks to be mined quickly.

Copy the following genesis definition to a file called privateNetworkGenesis.json and save it in the Private-Network directory:

{  
 "config": {  
 "constantinoplefixblock": 0,  
 "ethash": {  
 "fixeddifficulty": 1000  
 },  
 "chainID": 1981  
 },  
 "nonce": "0x42",  
 "gasLimit": "0x1000000",  
 "difficulty": "0x10000",  
 "alloc": {  
 "fe3b557e8fb62b89f4916b721be55ceb828dbd73": {  
 "privateKey": "8f2a55949038a9610f50fb23b5883af3b4ecb3c3bb792cbcefbd1542c692be63",  
 "comment": "private key and this comment are ignored. In a real chain, the private key should NOT be stored",  
 "balance": "0xad78ebc5ac6200000"  
 },  
 "f17f52151EbEF6C7334FAD080c5704D77216b732": {  
 "privateKey": "ae6ae8e5ccbfb04590405997ee2d52d2b330726137b875053c36d94e974d162f",  
 "comment": "private key and this comment are ignored. In a real chain, the private key should NOT be stored",  
 "balance": "90000000000000000000000"  
 }  
 }  
}

**Warning**

Do not use the accounts in the genesis file above on mainnet or any public network except for testing. The private keys are displayed so the accounts are not secure.

### 3. Get Public Key of First Node

To enable nodes to discover each other, a network requires one or more nodes to be bootnodes. For this private network, we will use Node-1 as the bootnode. This requires obtaining the public key for the [enode URL](https://docs.pantheon.pegasys.tech/en/stable/Configuring-Pantheon/Node-Keys/#enode-url).

In the Node-1 directory, use the [public-key subcommand](https://docs.pantheon.pegasys.tech/en/stable/Reference/Pantheon-CLI-Syntax/#public-key) to write the [node public key](https://docs.pantheon.pegasys.tech/en/stable/Configuring-Pantheon/Node-Keys/#node-public-key) to the specified file (publicKeyNode1 in this example):

**MacOS**

pantheon --data-path=Node-1-data-path --genesis-file=../privateNetworkGenesis.json public-key export --to=Node-1-data-path/publicKeyNode1

Windows

Your node 1 directory now contains:

├── Node-1  
 ├── Node-1-data-path  
 ├── database  
 ├── key  
 ├── publicKeyNode1

The database directory contains the blockchain data.

### 4. Start First Node as Bootnode

Start Node-1:

**MacOS**

pantheon --data-path=Node-1-data-path --genesis-file=../privateNetworkGenesis.json --bootnodes --miner-enabled --miner-coinbase fe3b557e8fb62b89f4916b721be55ceb828dbd73 --rpc-http-enabled --host-whitelist=\* --rpc-http-cors-origins="all"

Windows

The command line specifies:

* No arguments for the [--bootnodes](https://docs.pantheon.pegasys.tech/en/stable/Reference/Pantheon-CLI-Syntax/#bootnodes) option because this is your bootnode.
* Mining is enabled and the account to which mining rewards are paid using the [--miner-enabled](https://docs.pantheon.pegasys.tech/en/stable/Reference/Pantheon-CLI-Syntax/#miner-enabled) and [--miner-coinbase](https://docs.pantheon.pegasys.tech/en/stable/Reference/Pantheon-CLI-Syntax/#miner-coinbase) options.
* JSON-RPC API is enabled using the [--rpc-http-enabled](https://docs.pantheon.pegasys.tech/en/stable/Reference/Pantheon-CLI-Syntax/#rpc-http-enabled) option.
* All hosts can access the HTTP JSON-RPC API using the [--host-whitelist](https://docs.pantheon.pegasys.tech/en/stable/Reference/Pantheon-CLI-Syntax/#host-whitelist) option.
* All domains can access the node using the HTTP JSON-RPC API using the [--rpc-http-cors-origins](https://docs.pantheon.pegasys.tech/en/stable/Reference/Pantheon-CLI-Syntax/#rpc-http-cors-origins) option.

**Info**

The miner coinbase account is one of the accounts defined in the genesis file.

### 5. Start Node-2

You need the [enode URL](https://docs.pantheon.pegasys.tech/en/stable/Configuring-Pantheon/Node-Keys/#enode-url) for Node-1 to specify Node-1 as the bootnode for Node-2 and Node-3.

Start another terminal, change to the Node-2 directory and start Node-2 replacing the enode URL with your bootnode:

**MacOS**

pantheon --data-path=Node-2-data-path --genesis-file=../privateNetworkGenesis.json --bootnodes="enode://<node public key ex 0x>@127.0.0.1:30303" --p2p-port=30304

Windows

The command line specifies:

* Different port to Node-1 for P2P peer discovery using the [--p2p-port](https://docs.pantheon.pegasys.tech/en/stable/Reference/Pantheon-CLI-Syntax/#p2p-port) option.
* Enode URL for Node-1 using the [--bootnodes](https://docs.pantheon.pegasys.tech/en/stable/Reference/Pantheon-CLI-Syntax/#bootnodes) option.
* Data directory for Node-2 using the [--data-path](https://docs.pantheon.pegasys.tech/en/stable/Reference/Pantheon-CLI-Syntax/#data-path) option.
* Genesis file as for Node-1.

### 6. Start Node-3

Start another terminal, change to the Node-3 directory and start Node-3 replacing the enode URL with your bootnode:

**MacOS**

pantheon --data-path=Node-3-data-path --genesis-file=../privateNetworkGenesis.json --bootnodes="enode://<node public key ex 0x>@127.0.0.1:30303" --p2p-port30305

Windows

The command line specifies:

* Different port to Node-1 and Node-2 for P2P peer discovery.
* Data directory for Node-3 using the [--data-path](https://docs.pantheon.pegasys.tech/en/stable/Reference/Pantheon-CLI-Syntax/#data-path) option.
* Bootnode and genesis file as for Node-2.

### 7. Confirm Private Network is Working

Start another terminal, use curl to call the JSON-RPC API [net\_peerCount](https://docs.pantheon.pegasys.tech/en/stable/Reference/JSON-RPC-API-Methods/#net_peercount) method and confirm the nodes are functioning as peers:

curl -X POST --data '{"jsonrpc":"2.0","method":"net\_peerCount","params":[],"id":1}' 127.0.0.1:8545

The result confirms Node-1 (the node running the JSON-RPC service) has two peers (Node-2 and Node-3):

{  
 "jsonrpc" : "2.0",  
 "id" : 1,  
 "result" : "0x2"  
}

## Next Steps

Import accounts to MetaMask and send transactions as described in the [Private Network Quickstart Tutorial](https://docs.pantheon.pegasys.tech/en/stable/Tutorials/Private-Network-Quickstart/#creating-a-transaction-using-metamask)

**Info**

Pantheon does not implement [private key management](https://docs.pantheon.pegasys.tech/en/stable/Using-Pantheon/Account-Management/).

Send transactions using eth\_sendRawTransaction to [send ether or, deploy or invoke contracts](https://docs.pantheon.pegasys.tech/en/stable/Using-Pantheon/Transactions/).

Use the [JSON-RPC API](https://docs.pantheon.pegasys.tech/en/stable/JSON-RPC-API/Using-JSON-RPC-API/).

Start a node with the [--rpc-ws-enabled](https://docs.pantheon.pegasys.tech/en/stable/Reference/Pantheon-CLI-Syntax/#rpc-ws-enabled) option and use the [RPC Pub/Sub API](https://docs.pantheon.pegasys.tech/en/stable/Using-Pantheon/RPC-PubSub/).

## Stop Nodes

When finished using the private network, stop all nodes using Ctrl+C in each terminal window.

**Tip**

To restart the private network in the future, start from [4. Restart First Node as Bootnode](https://docs.pantheon.pegasys.tech/en/stable/Tutorials/Create-Private-Network/#4-restart-first-node-as-bootnode).