Proof of Authority Development Chain

For this assignment, you will take on the role of a new developer at a small bank.

Your mission, should you choose to accept it, will be to set up a testnet blockchain for your organization.

To do this, you will create and submit four deliverables:

- Set up your custom testnet blockchain.
- Send a test transaction.
- Create a repository.
- Write instructions on how to use the chain for the rest of your team.

Background

You have just landed a new job at ZBank, a small, innovative bank that is interested in exploring what blockchain technology can do for them and their customers.

Your first project at the company is to set up a private testnet that you and your team of developers can use to explore potentials for blockchain at ZBank.

You have decided on setting up a testnet because:

There is no real money involved, which will give your team of developers the freedom to experiment.

Testnets allows for offline development.

In order to set up a testnet, you will need to use the following skills/tools we learned in class:

- Puppeth, to generate your genesis block.
- Geth, a command-line tool, to create keys, initialize nodes, and connect the nodes together.
- The Clique Proof of Authority algorithm.

Tokens inherently have no value here, so we will provide pre-configured accounts and nodes for easy setup.

After creating the custom development chain, create documentation for others on how to start it using the pre-configured nodes and accounts. You can name the network anything you want, have fun with it!

Be sure to include any preliminary setup information, such as installing dependencies and environment configuration.

Instructions

Setup the custom out-of-the-box blockchain

- Create a new project directory for your new network. Call it whatever you want!
- Create a "Screenshots" folder inside of the project directory.

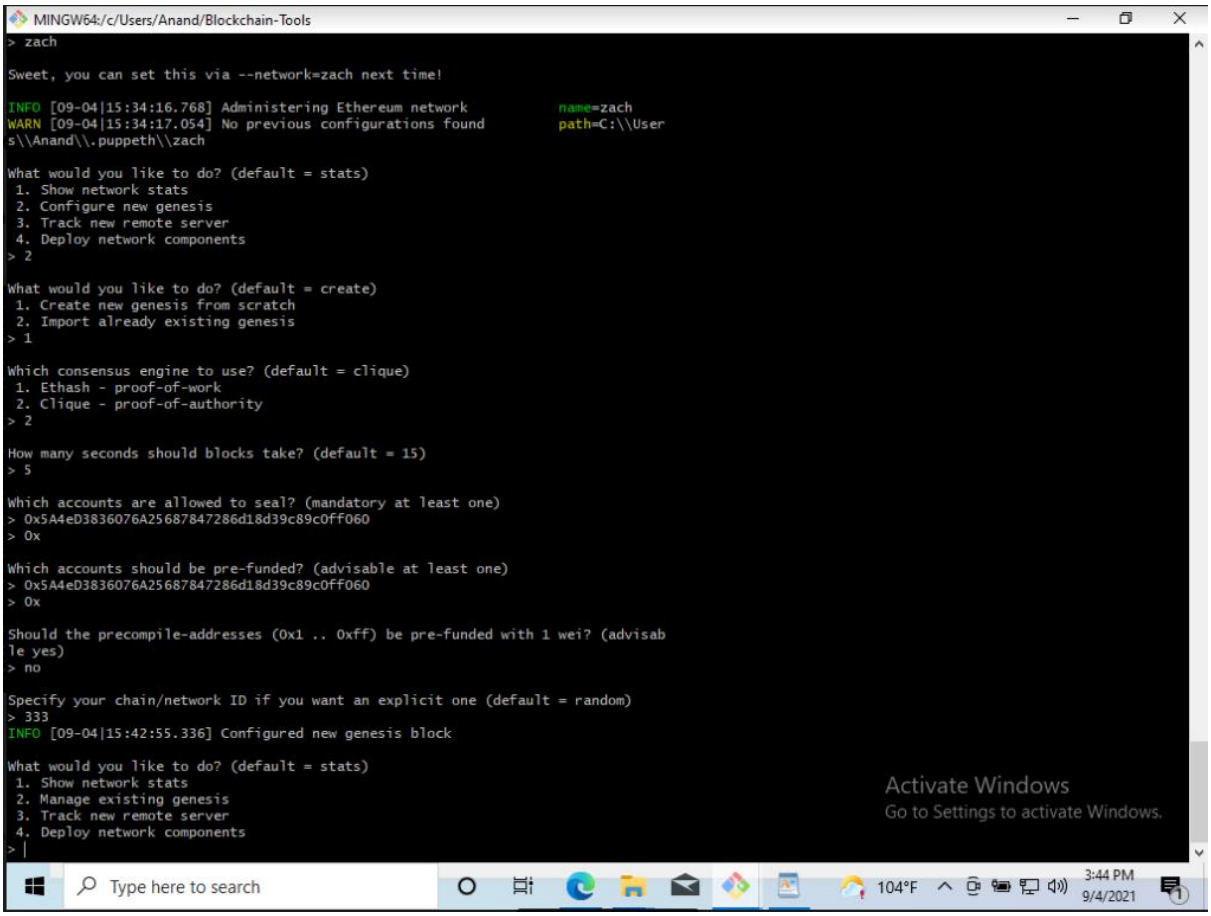
This directory was created as Blockchain Tools and contains all the needed commands to create and run the network.

- Create accounts for two (or more) nodes for the network with a separate `datadir` for each using `geth`.

Two nodes were created using geth and were called node1 and node2. They

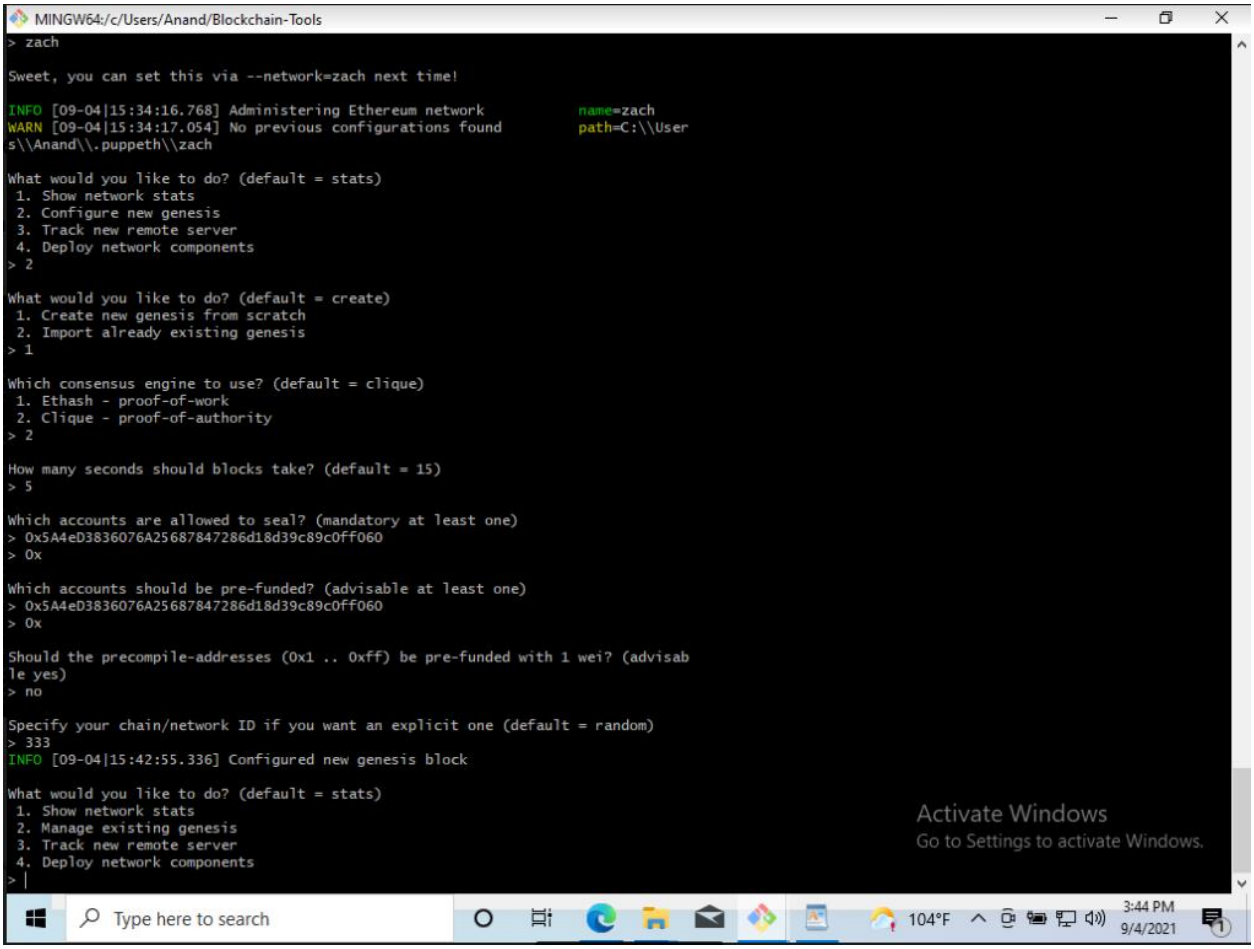
- Run `puppeth`, name your network, and select the option to configure a new genesis block.
- Choose the `Clique (Proof of Authority)` consensus algorithm.

This is what was done for the proof of authority development chain running the puppeth command.



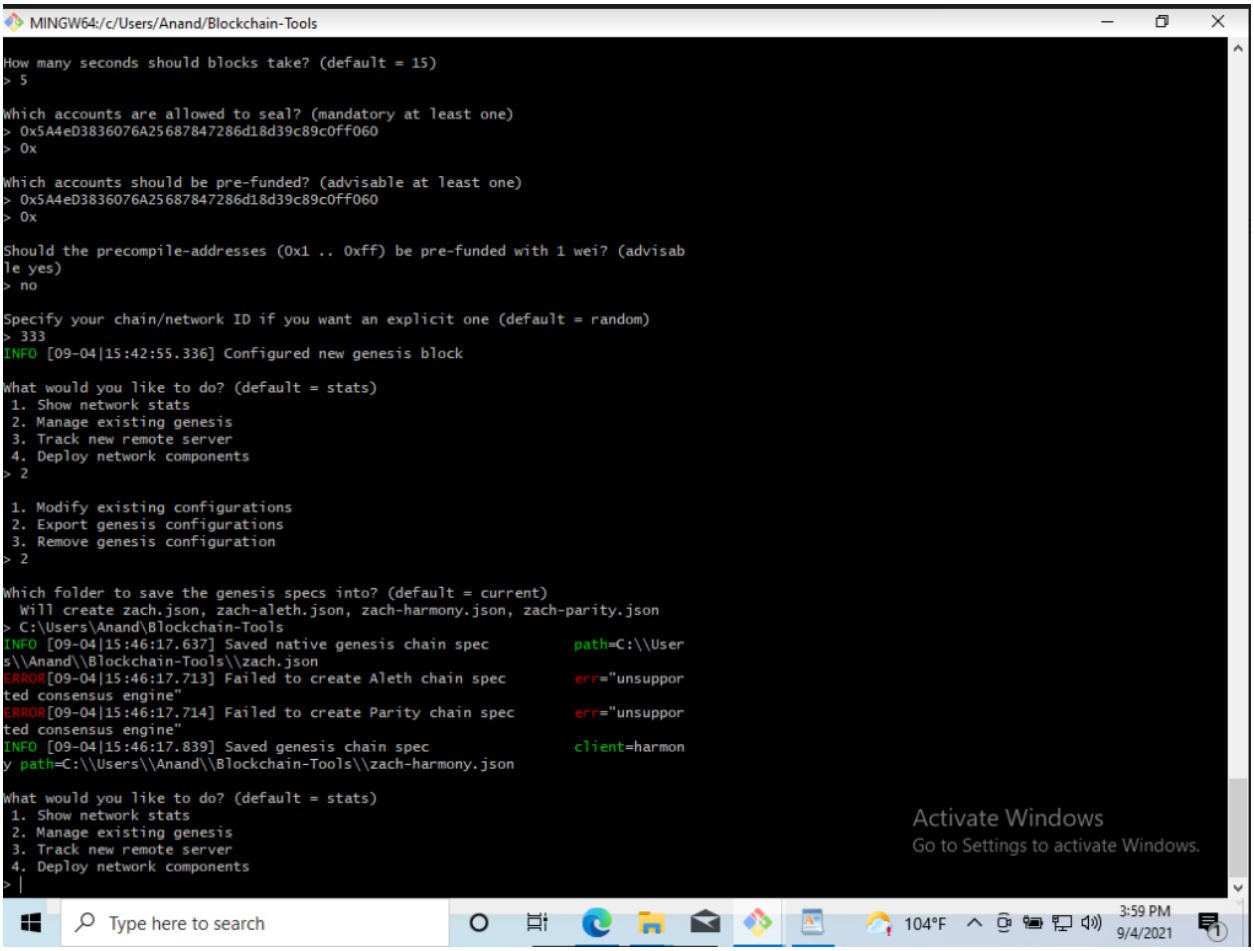
- Paste both account addresses from the first step one at a time into the list of accounts to seal.
- Paste them again in the list of accounts to pre-fund. There are no block rewards in PoA, so you'll need to pre-fund.
- You can choose `no` for pre-funding the pre-compiled accounts (0x1 .. 0xff) with wei. This keeps the genesis cleaner.
- Complete the rest of the prompts, and when you are back at the main menu, choose the "Manage existing genesis" option.

Further steps on the puppet command to create the network and manage the existing genesis.



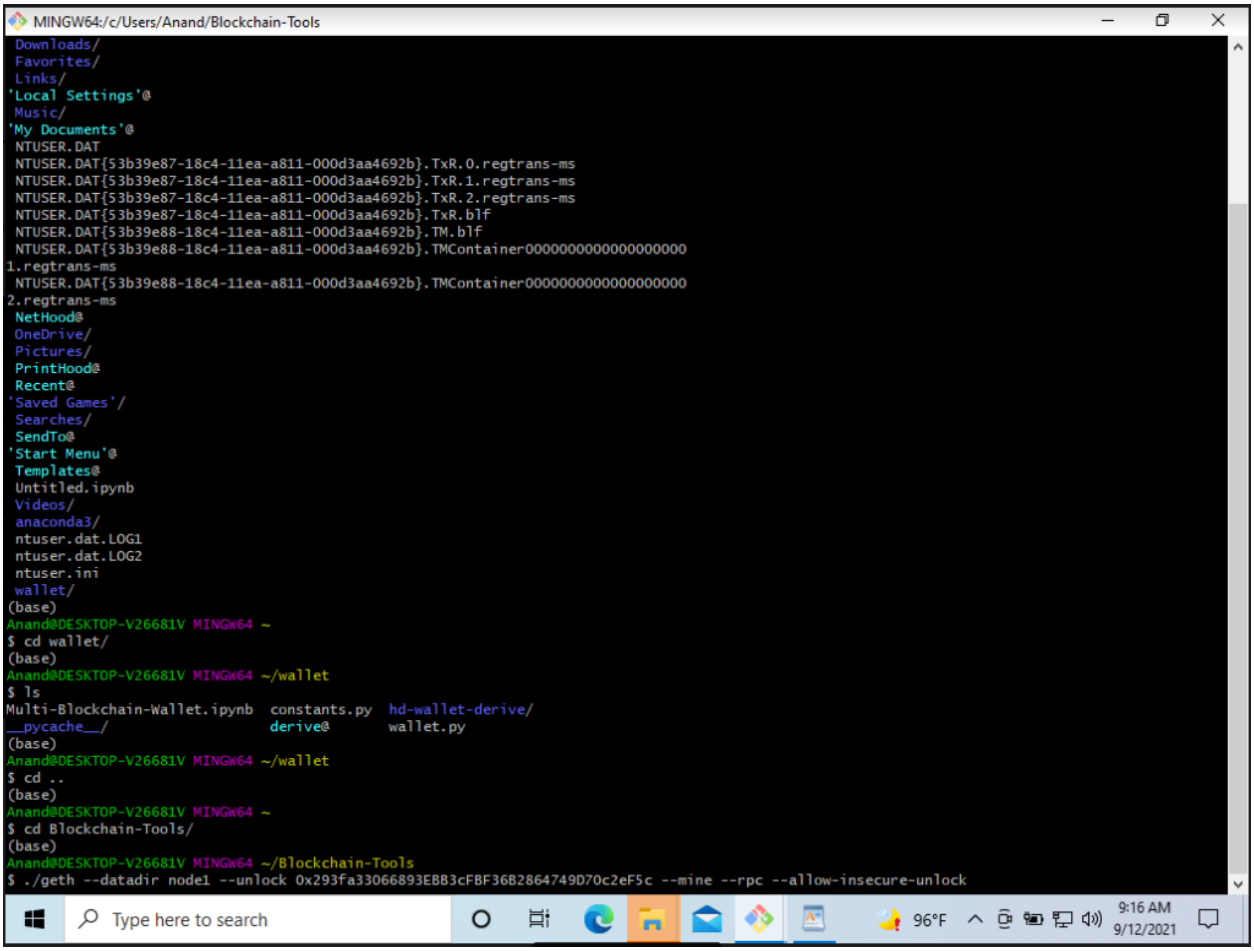
- Export genesis configurations. This will fail to create two of the files, but you only need `networkname.json`.
- You can delete the `networkname-harmony.json` file.
- Screenshot the `puppeth` configuration once complete and save it to the Screenshots folder.

Network was named zach after our instructor:

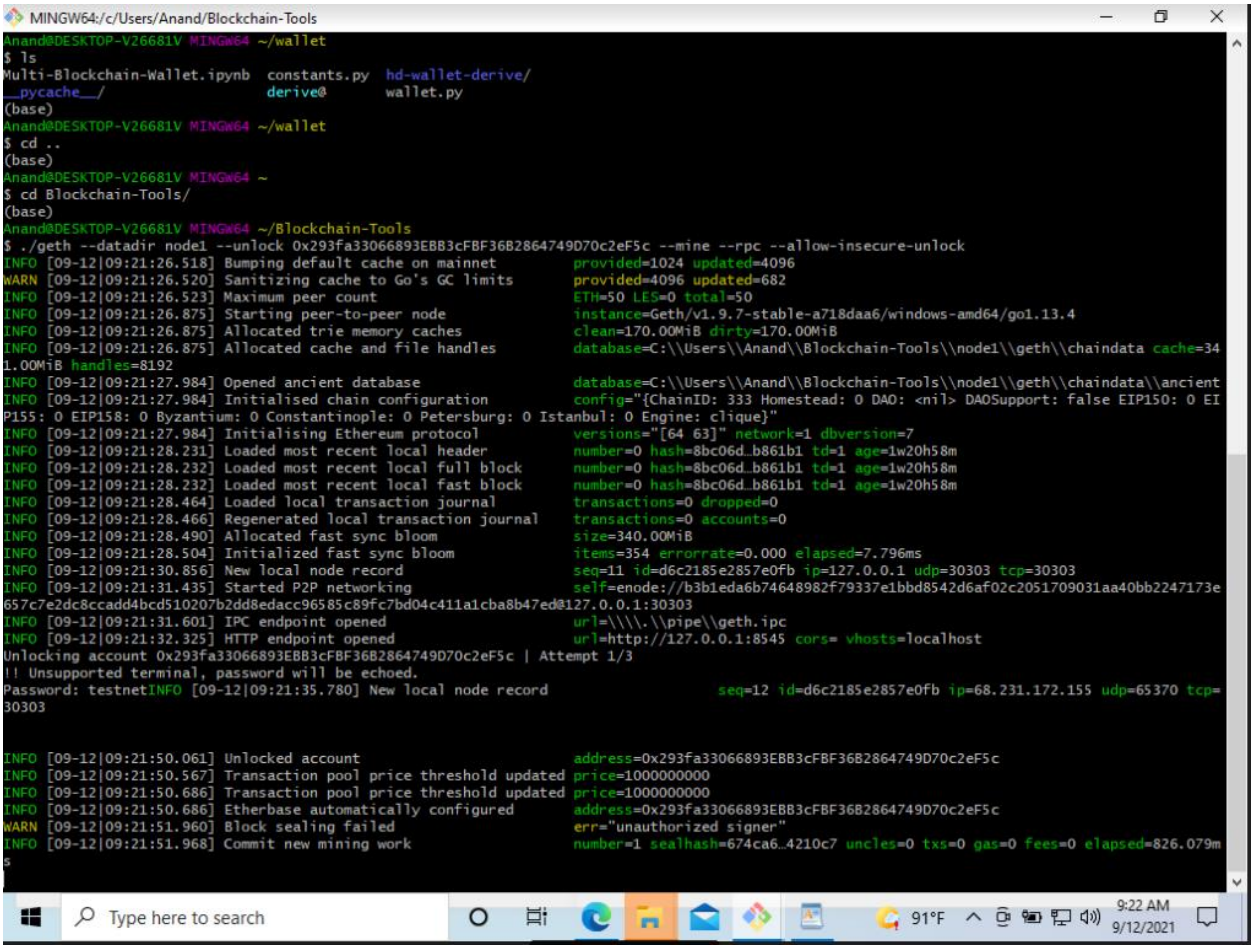


- Initialize each node with the new `networkname.json` with `geth`.
- Run the first node, unlock the account, enable mining, and the RPC flag. Only one node needs RPC enabled.
- Set a different peer port for the second node and use the first node's `enode` address as the `bootnode` flag.
- Be sure to unlock the account and enable mining on the second node!
- You should now see both nodes producing new blocks, congratulations!

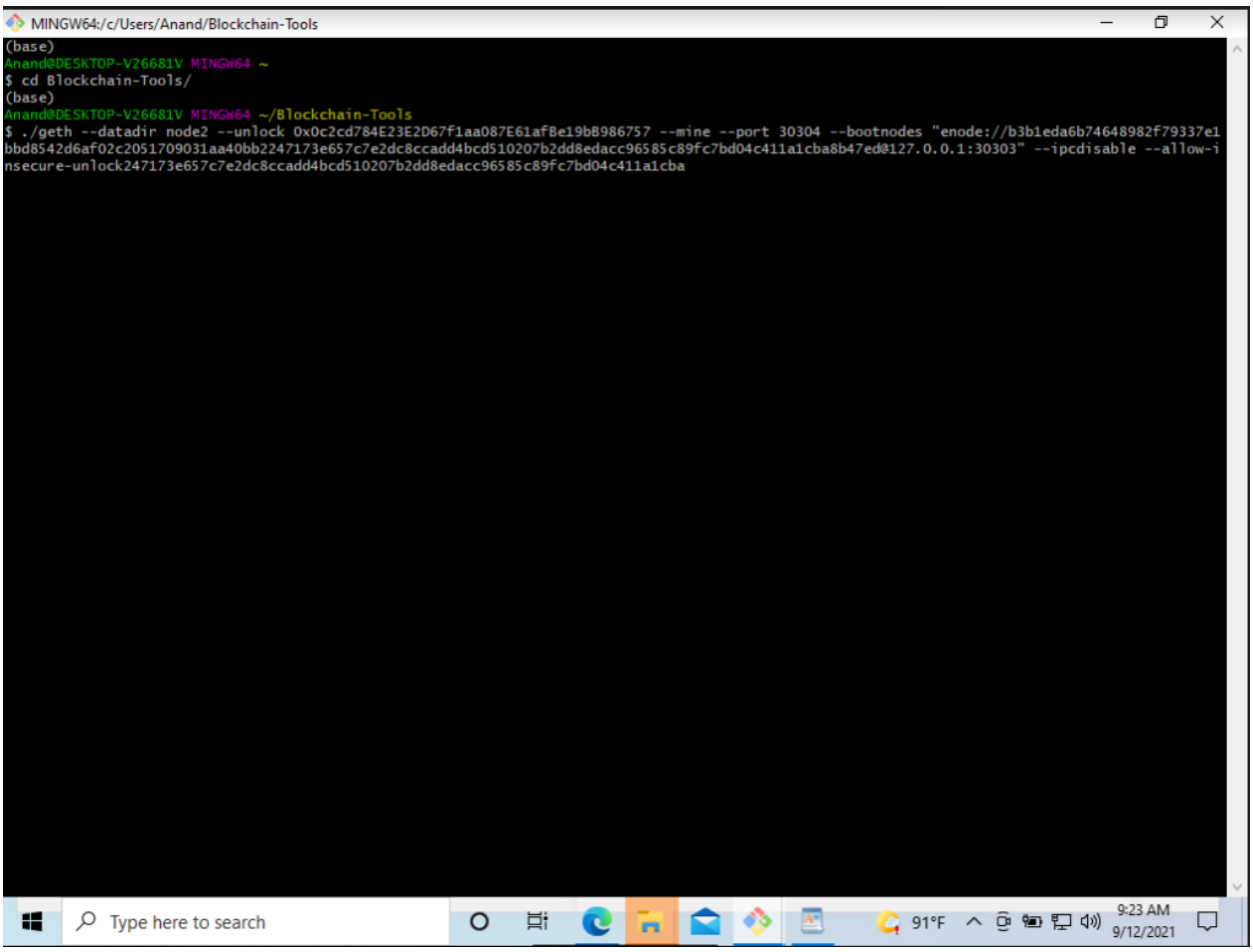
Using geth, I ran the two nodes. Here is the command to run node1:



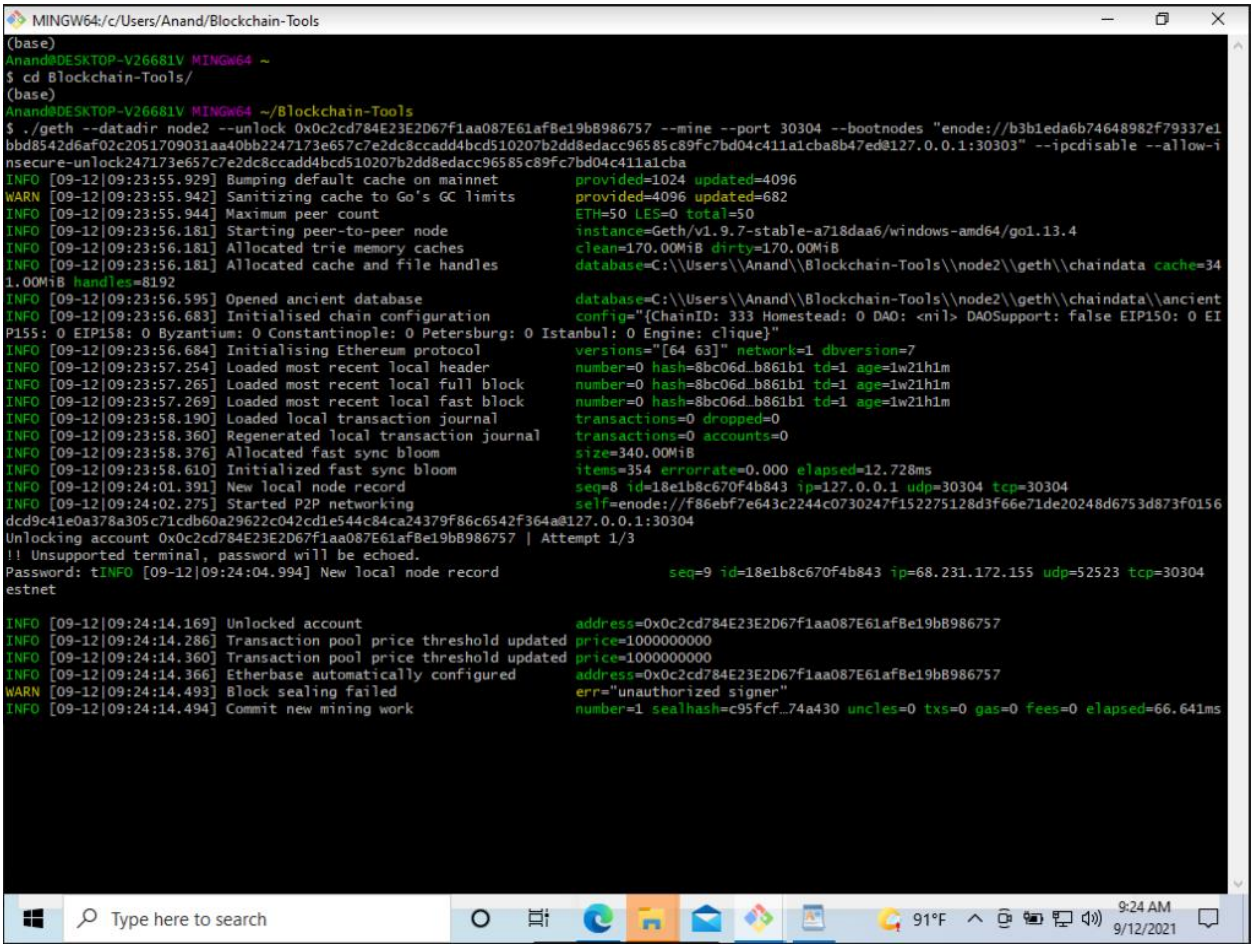
This is the result of the command. I put in the testnet password to unlock the node.



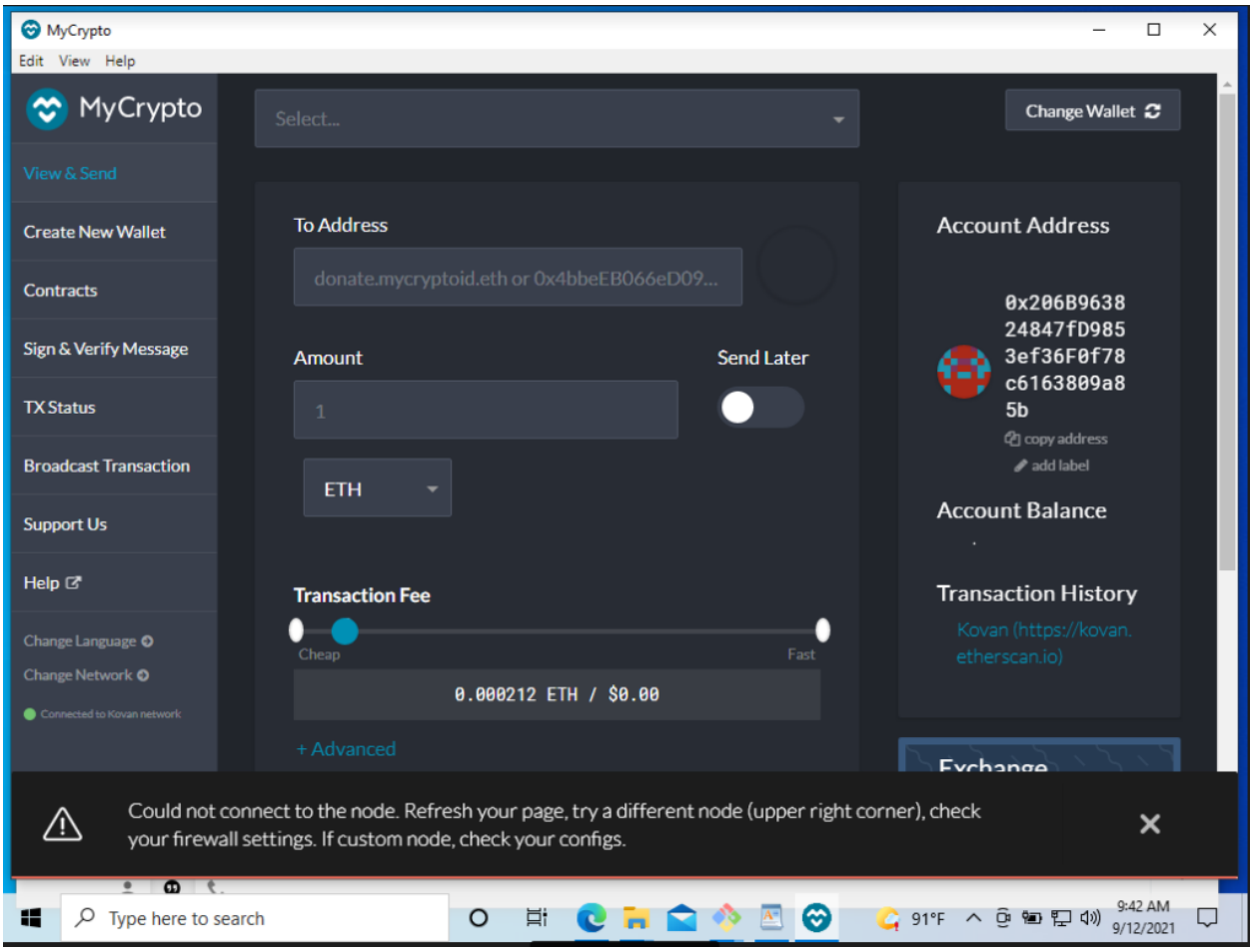
Then I opened up another terminal and ran the command for node2:



This is the result along with the password to unlock the node:



The nodes were both running. This is where I ran into problems. I tried connecting to the network and was not able to. I checked the firewall and the settings, but all of those seemed to be ok. I tried running it in a different machine and got the same result. Here is one of several attempts to connect to the zach network including creating a new zach1 network. None of these attempts bore fruit. For future work, I will look into this and maybe use some software other than mycrypto to see if it will connect:


MyCrypto
Edit View Help
MyCrypto
Select...
Change Wallet
View & Send
Create New Wallet
Contracts
Sign & Verify Message
TX Status
Broadcast Transaction
Support Us
Help
Change Language
Change Network
Connected to Kovan network
To Address
donate.mycryptoid.eth or 0x4bbeEB066eD09...
Amount
Send Later
1
ETH
Transaction Fee
Cheap
Fast
0.000212 ETH / $0.00
+ Advanced
Account Address
0x206B963824847fD9853ef36F0f78c6163809a85b
copy address
add label
Account Balance
Transaction History
Kovan (https://kovan.etherscan.io)
Exchange
Could not connect to the node. Refresh your page, try a different node (upper right corner), check your firewall settings. If custom node, check your configs.
Type here to search
91°F
9:42 AM
9/12/2021