Introduction to blockchain and Ethereum smart contracts

Create your own Ethereum private network

Written by Peter, Ho Man Fai

Microsoft

The Hong Kong Polytechnic University
香港理工大學

Department of Computing
電子計算學系

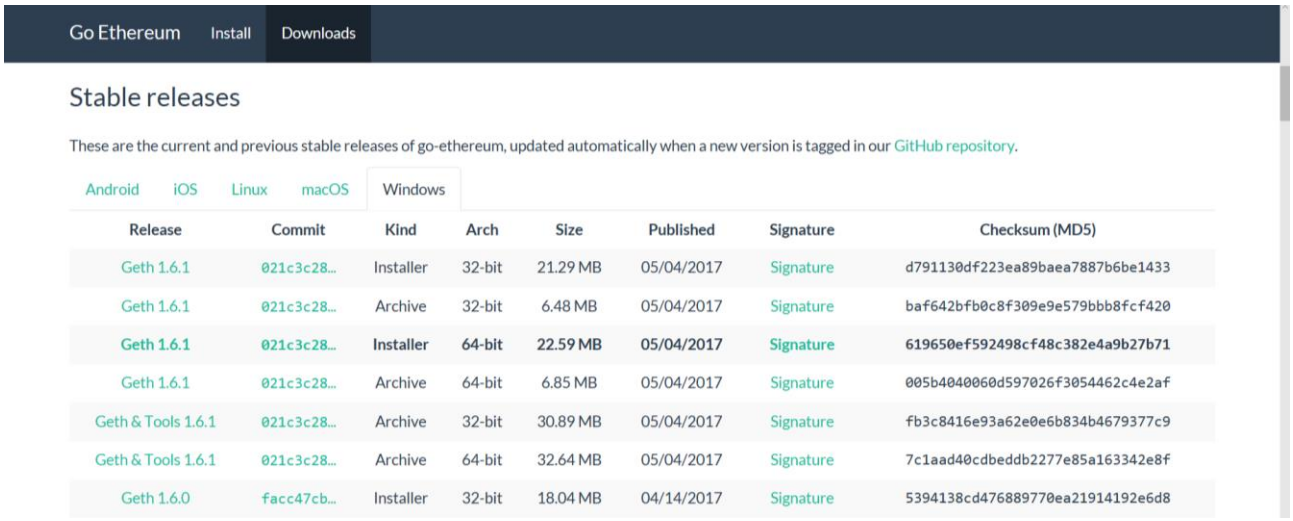
Learning outcomes

After you complete this lab, you will be able to:

1. Bootstrap a new Ethereum blockchain using Geth.
2. Understand the usage of the genesis.json and static-nodes.json files.
3. Run an Ethereum node on a computer.
4. Connect an Ethereum node to another node on the same network.

Part 1: Install Ethereum implementation: Geth (Go Ethereum)

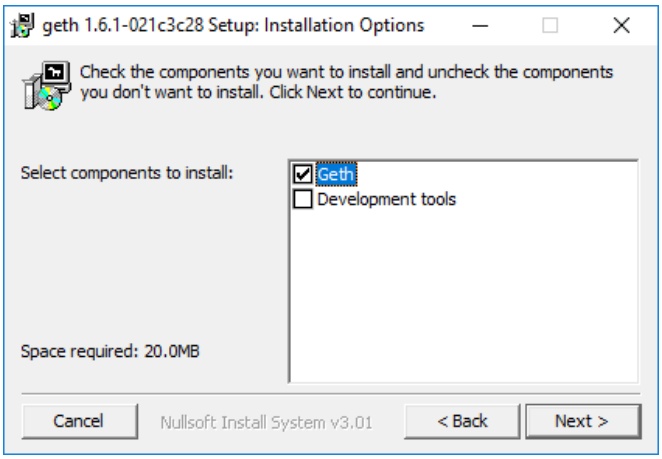
1. Visit https://geth.ethereum.org/downloads/ → Scroll down and look for the "Stable releases" section → Click the Windows tab → Click "**Geth 1.7.2 Installer**" row on the table.



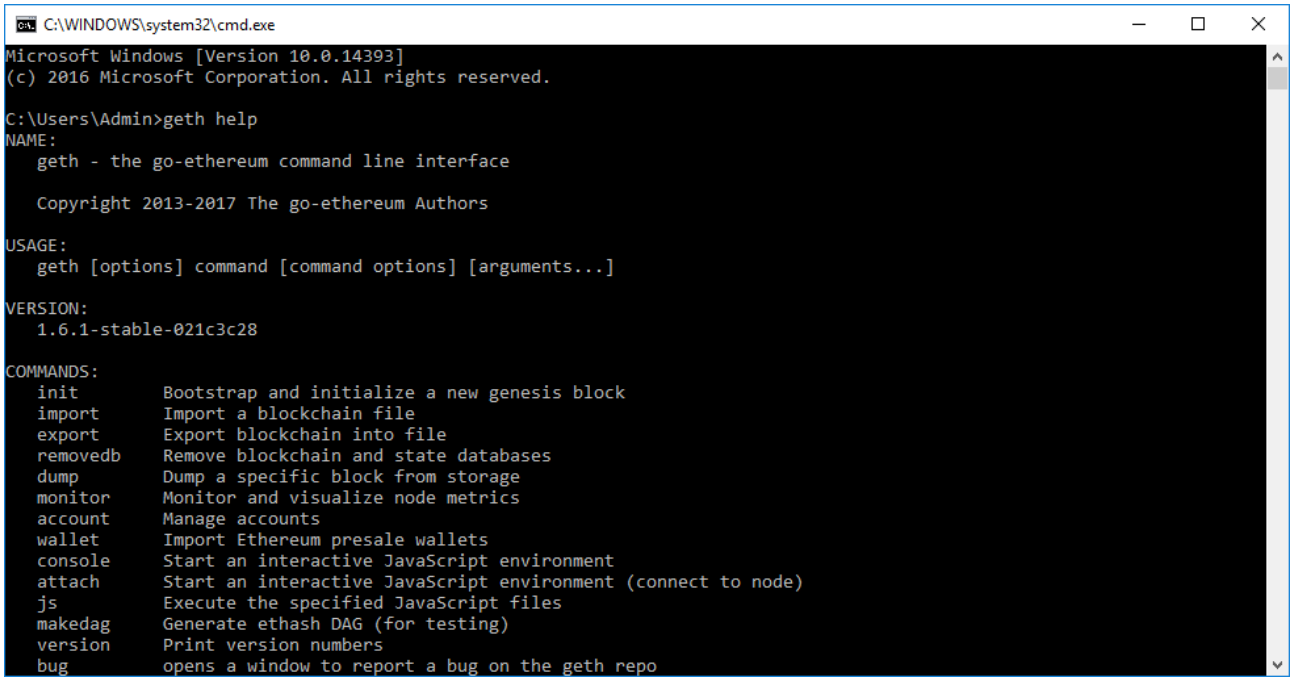
Important note:

The latest version of Geth keeps updating. The developers of Geth will introduce and deprecate some of the features as time goes by. By the time you are reading this, you should be able to download a version that is beyond 1.7.2. However, DO NOT use other versions except 1.7.2 for this series of tutorial. It is because this tutorial is written for version 1.7.2. We do not guarantee the materials here will work other than that version.

2. Download the installer to your desktop → Click the installer for installation (no need to check the "Development tools" box when selecting installation options).

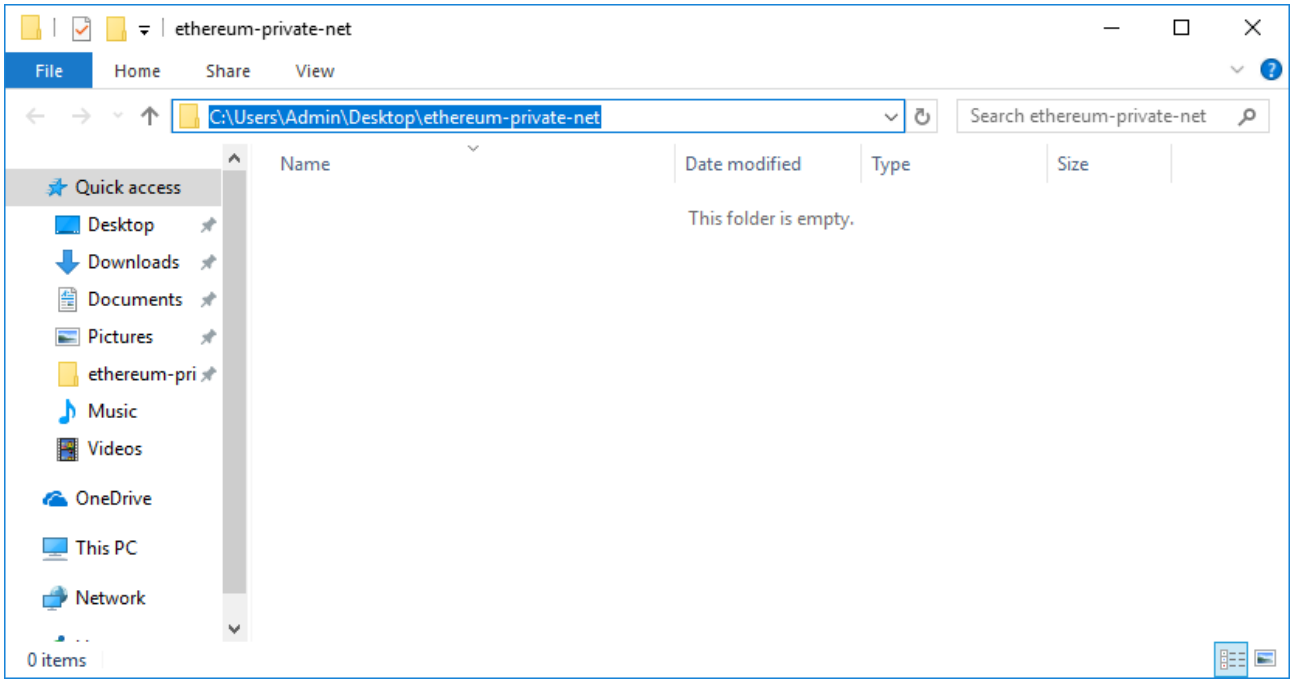


3. Open command prompt → run "geth help" → If you see a list of help instructions being generated, that means you have successfully installed the Geth implementation.

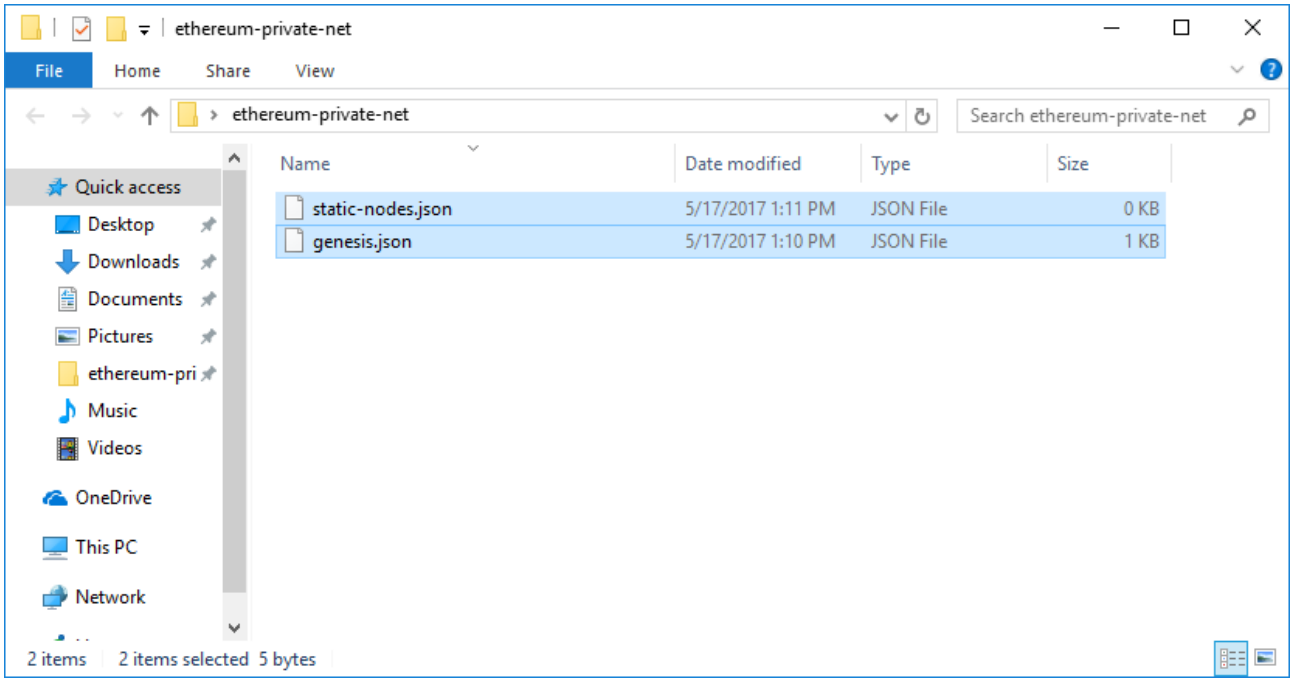


Part 2: Initialize a new private Ethereum blockchain data

4. Go to your Windows desktop (or other directories if you like) → **Create a new folder called "ethereum-private-net"** (or any name you want). It is recommended not to include any spaces in your path to the folder directory.



5. Open your text editor (e.g. Notepad++, but DO NOT use the Windows default notepad.exe) → **Create a JSON file named as "genesis.json" and another JSON file named as "static-nodes.json"** inside the folder you have just created in the previous step.



6. Edit your genesis.json → Paste the following JSON content inside → Hit save button.

```
{
    "config": {
        "chainId": 15,
        "homesteadBlock": 0,
        "eip155Block": 0,
        "eip158Block": 0
    },
    "difficulty": "10000",
    "gasLimit": "2100000",
    "alloc": {}
}
```

7. Edit your static-nodes.json → Paste the following JSON content inside → Hit save button.

```
[]
```

Explanations:

Blockchain, as the name suggests, is consisted of a chain of blocks. For the blockchain to work, we need to have the first and the initial block being defined (also known as the bootstrapping process), and there is a special name for such a block, called genesis block.

That's where the genesis.json file comes in. This file defines all the information and initial settings of your blockchain. Without defining this block, your blockchain will not know where and what to start with.

The explanations of those parameters inside the genesis.json is out of the scope of this tutorial. If you wish to learn more, check out the official documentation on Ethereum website.

Another file is the static-nodes.json. Please be reminded that Ethereum is a peer-to-peer blockchain network, and it means a participant (called node) needs to have direct connections between different participants (i.e. node). So the question is, who is/are the first one that a node should look for? That's where the static-nodes.json comes into play.

static-nodes.json is used to define all Ethereum addresses that will be connected to when you start your blockchain node in your computer. Since we haven't setup another computer yet, we leave the JSON an empty array, and we will come back and modify it later.

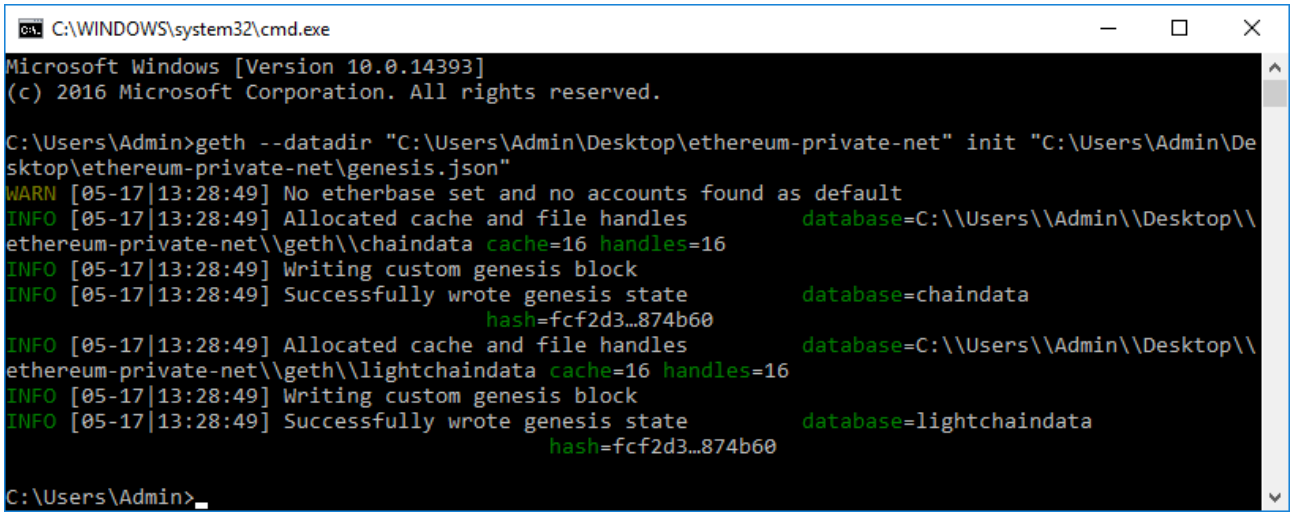
8. Open command prompt → **Run the following command**, you may need to change the highlighted paths below to match the path of your folder created.

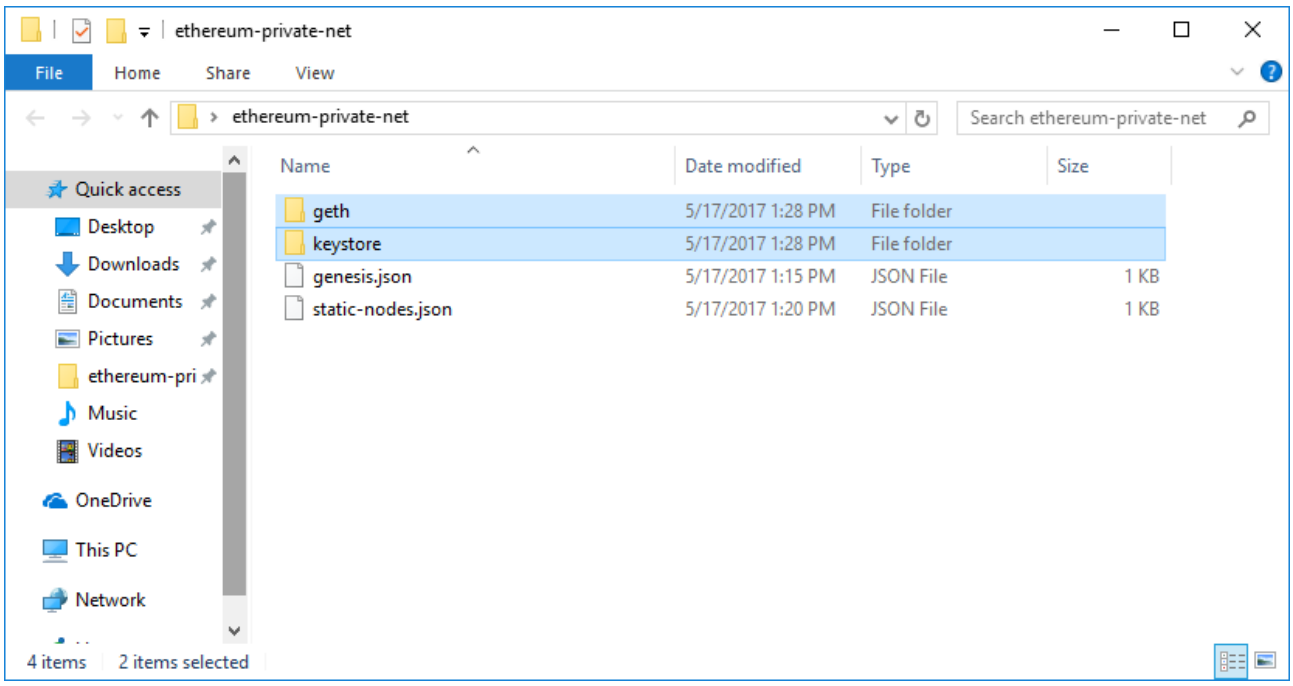
```
geth --datadir "C:\Users\Admin\Desktop\ethereum-private-net" init
"C:\Users\Admin\Desktop\ethereum-private-net\genesis.json"
```

Important note:

Try not to copy commands directly from this document since the quote characters may have changed to some other invalid characters when you paste it into the command prompt. **Type in the command by yourself.**

9. When done, you should see something like below in the command prompt, and you will notice there are two folders being generated inside your ethereum-private-net folder.





Explanations:

The geth folder stores all the blockchain data. The keystore folder stores all the keys of your Ethereum accounts. We will learn more about Ethereum accounts in the future.

Part 3: Start a Ethereum node

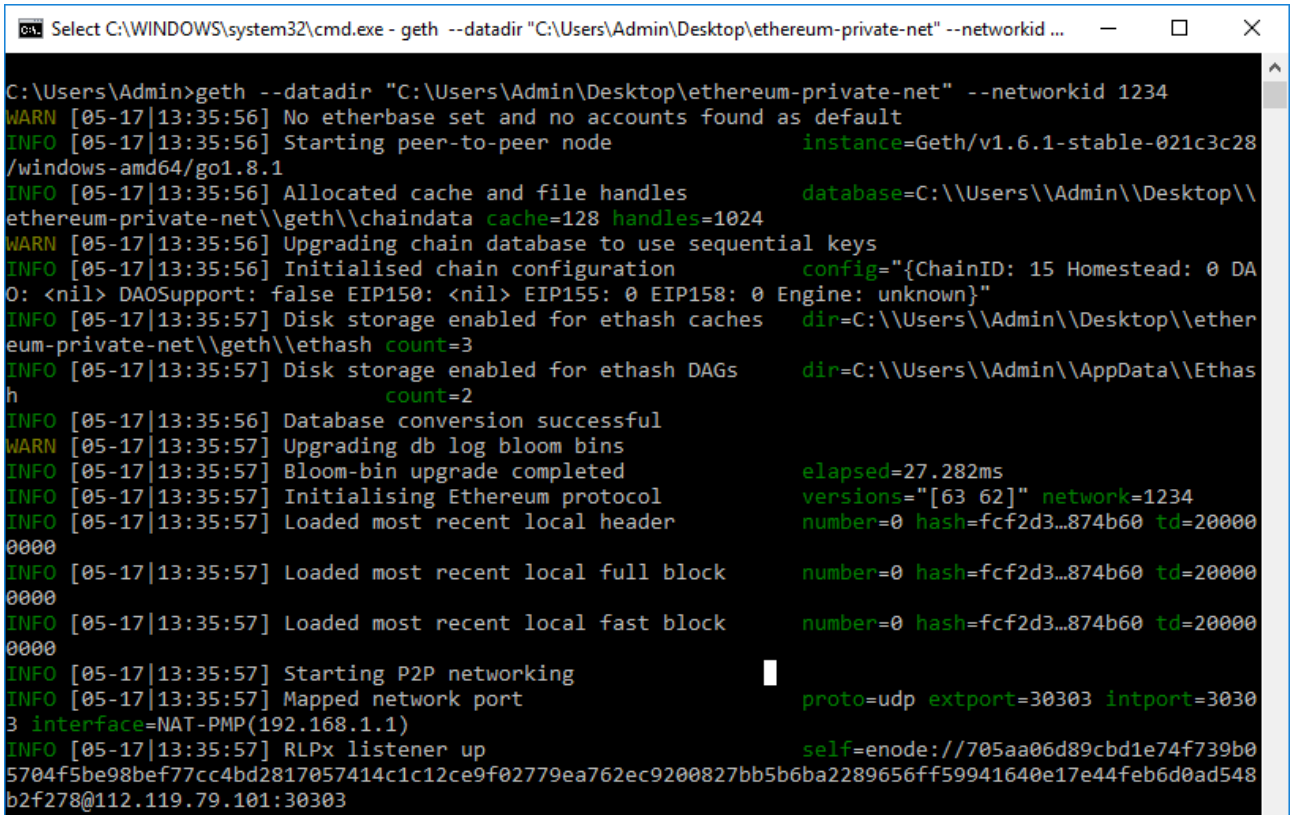
10. In the command prompt, **run the following command**. **You should change the green-color highlighted number into an arbitrary number**, you may also need to change the yellow-highlighted paths below to match the path of your folder created.

```
geth --datadir "C:\Users\Admin\Desktop\ethereum-private-net" --
networkid 1234567890
```

Important note:

This network ID should be unique in your network. For example, if you and your friend are doing your work in a computer lab that shares the same network, both of you should use different network IDs. Otherwise, the blockchain may generate conflicts and unexpected results when both of you running your own blockchain. So, **try to use a true random number instead of 1234567890.**

11. When done, you should see something like below. **This is what we called a Ethereum node daemon.**



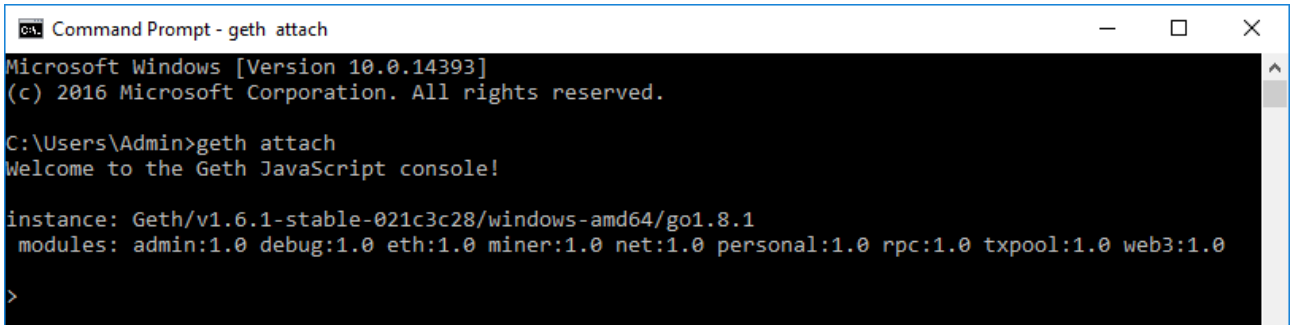
Important note:

The Ethereum node is running in your first computer now and it needs to keep running. So, **DO NOT terminate the command prompt or kill the process.**

Part 4: Create and connect to a second Ethereum node

12. Switch on your second computer or virtual machine → **repeat all the steps in part 1 and part 2 (no need to complete step 3 at this moment) on your second computer**.

13. Go back to your first computer → **Open a new command prompt** → **Run "geth attach"** → You should have entered the Ethereum interactive console like the image below.



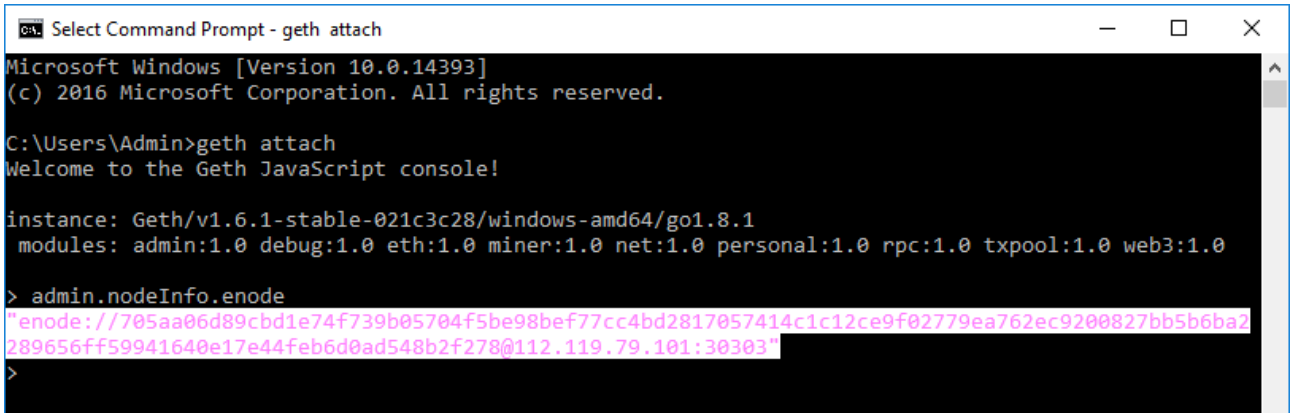
Important note:

Make sure your Ethereum node is running in another command prompt. If you have terminated the Ethereum node, you will not be able to enter the interactive console.

14. Inside the interactive console, run the following code.

```
admin.nodeInfo.enode
```

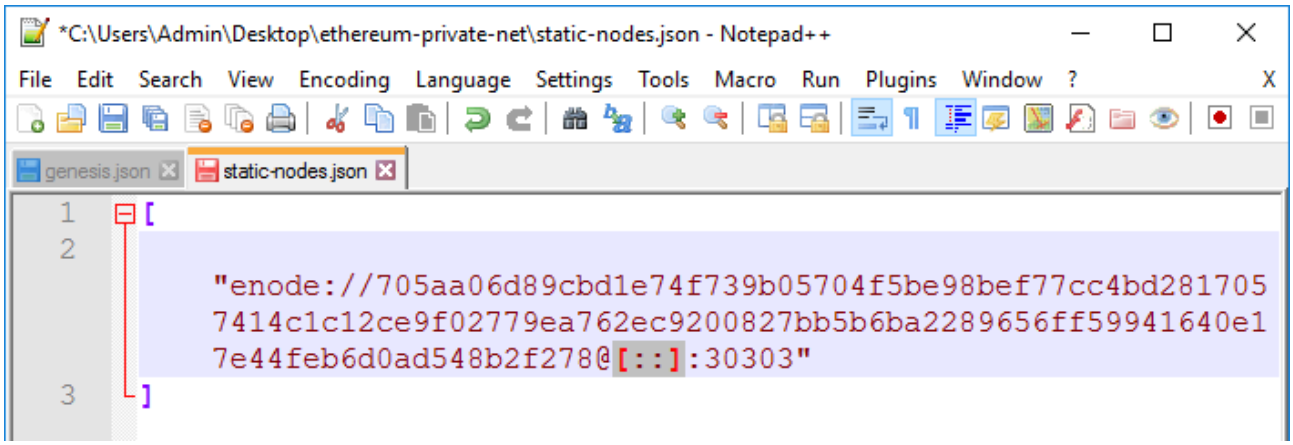
15. The console will print out an URL-like address of the current Ethereum node → Copy the address printed.



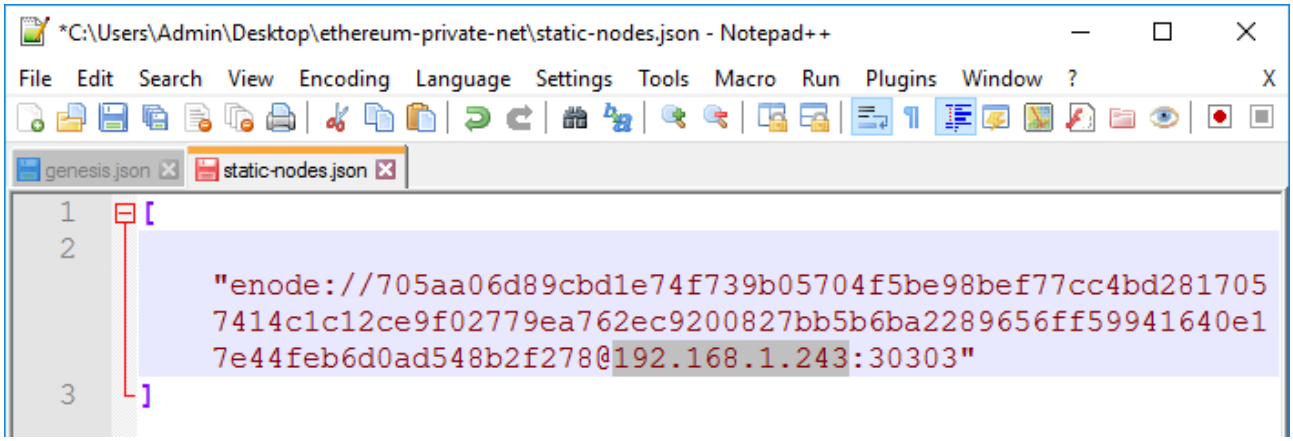
Explanations:

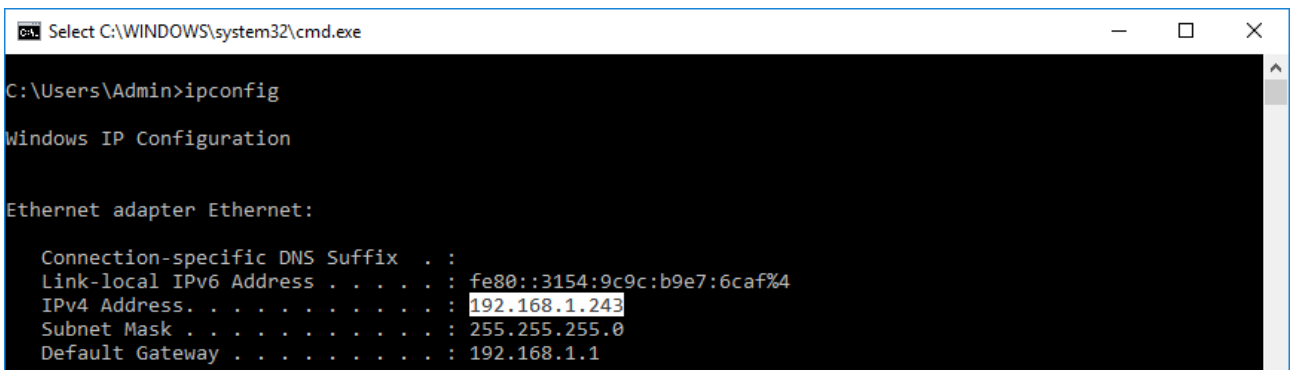
This URL-like address is the address string of your Ethereum node that you are running. This string is unique across the world (theoretically). There is also an IP address and port number that states where your Ethereum node resides on your network.

16. Go back to your second computer / virtual machine → **Open the static-nodes.json inside your ethereum-private-net folder → paste the address string copied from the first computer** inside the square bracket (see below).



17. Notice that there is an @ sign within the string, and you should see an IP address or [::] after the @ sign → **Replace the IP address or [::] with the first computer's internal network IP address**, you may find the IP address by running "ipconfig" in the command prompt of the first computer → save the file.





Important note:
There is a port number followed by the IP address (i.e. :30303). **DO NOT** modify it. You must keep it for the Ethereum nodes to connect via the 30303 port.

18. **Repeat all the steps stated in part 3 in your second computer** for starting the Ethereum node.

19. In your second computer, **open a new command prompt → Run "geth attach"** to start the Ethereum interactive console.

20. In the console opened, **run the following code** to see how many Ethereum node it has connected to.

```
net.peerCount
```

21. **You should see the output "1" in your console**. It means your Ethereum node inside the second computer has connected to an another Ethereum node on the network, which is your first computer.

22. **Go back to your first computer → Run the same command in the interactive console →** You should also see the output is also "1", meaning the connection is bi-directional.

Now, both of your Ethereum nodes are connected. However, if you start your second Ethereum node first next time, and then the first one, the connection will not be established. It is because you haven't defined your first computer's Ethereum node address in your second computer.

23. In your second computer, repeat step 14 – 17: copy the Ethereum address from your second computer → paste that address to the static-nodes.json file in your first computer (remember to replace the IP address or [::] string in the Ethereum address to be the internal IP address of the second computer).