Dhanashree Srinivasa

SUID: 393473169

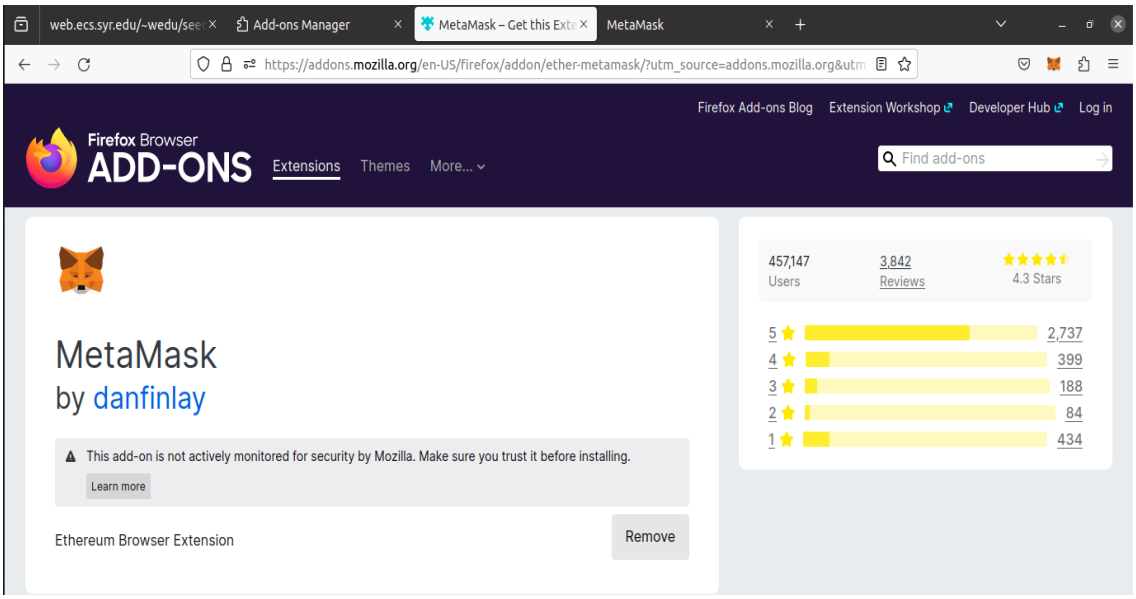
Course: Computer Security - CSE 643

Blockchain Exploration Lab

Task 1: Setting Up MetaMask Wallet

Task 1.a. Installing the MetaMask extension

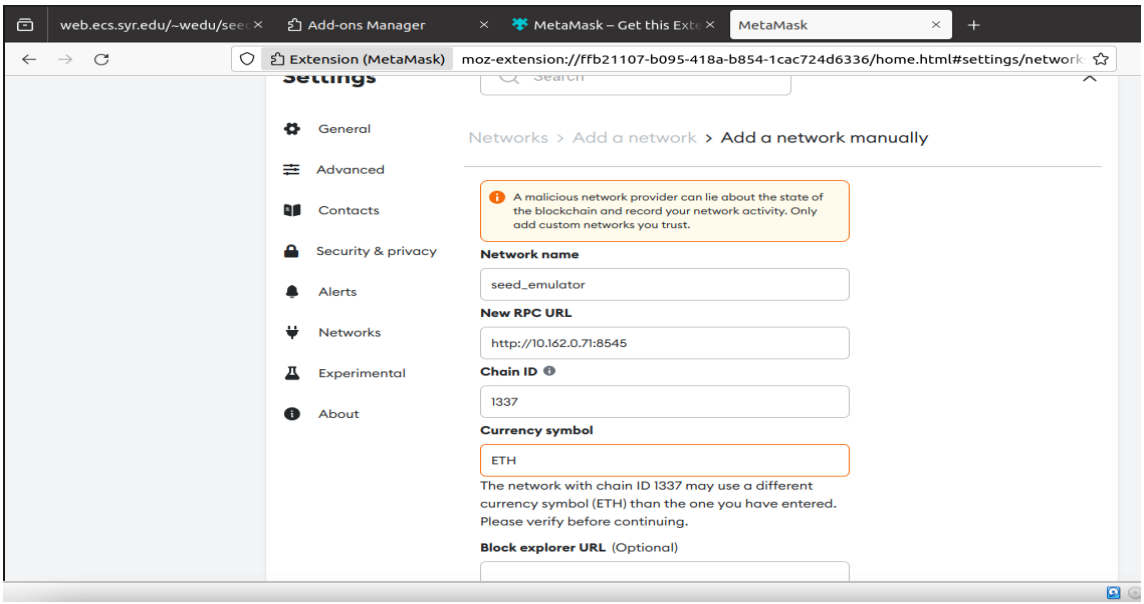
Open the firefox Add-ons and add the Metamask extension on firefox.



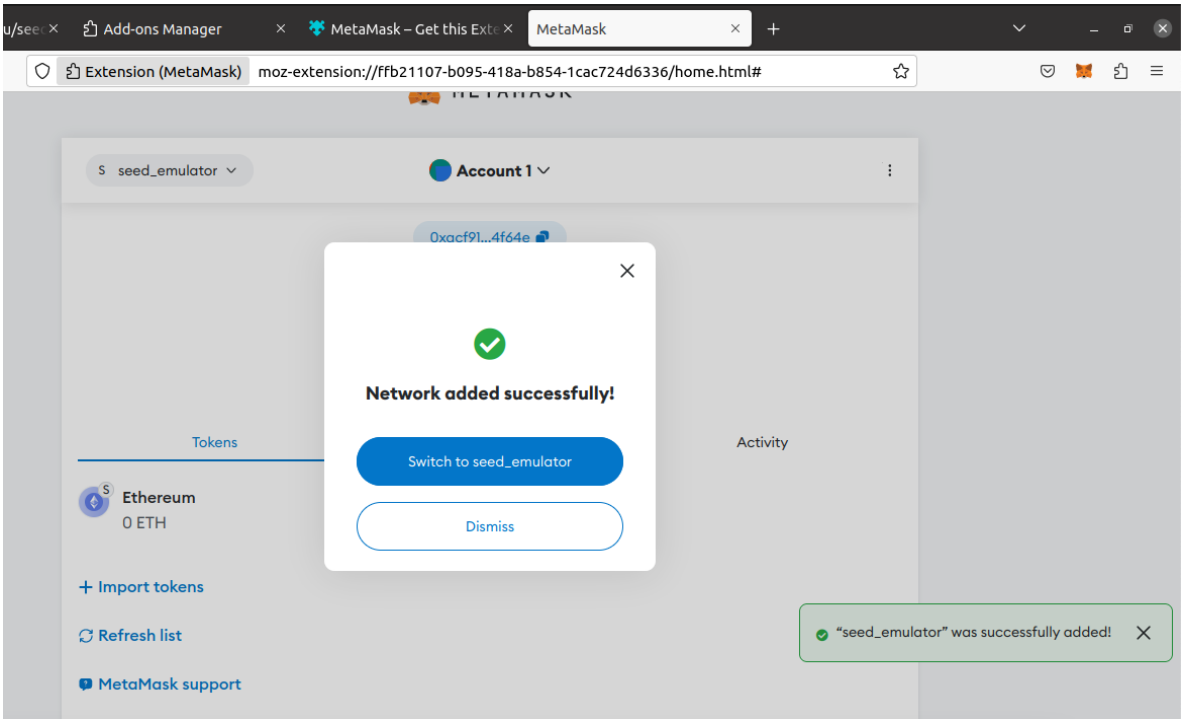
Task 1.b. Connecting to the Blockchain

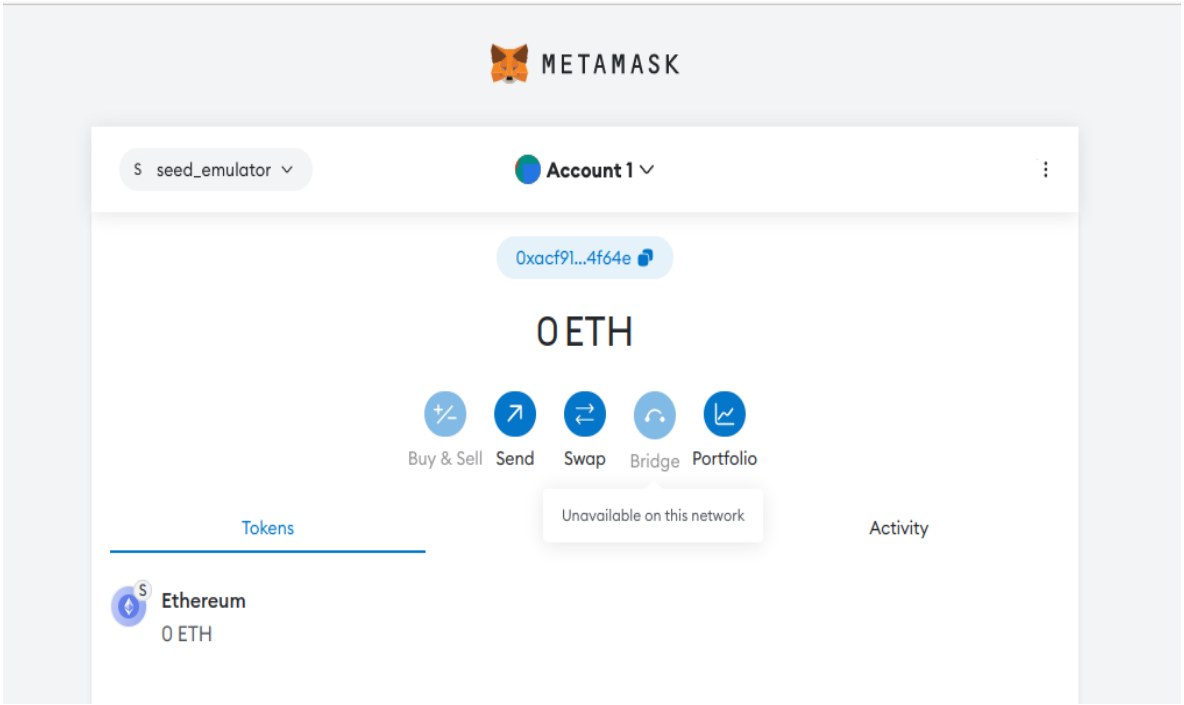
Here we must connect to Metamask to blockchain, for that get the ip address of a container.

Navigate to the settings menu on metamask and set the RPC URL.



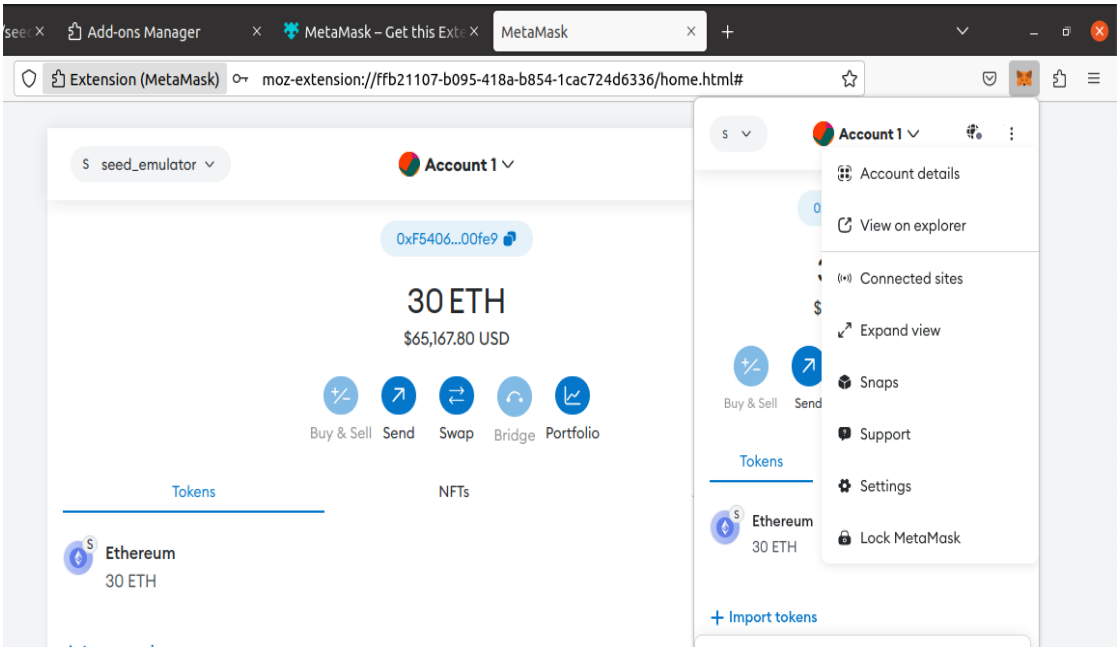
We can see that seed_emulator network is added successfully.

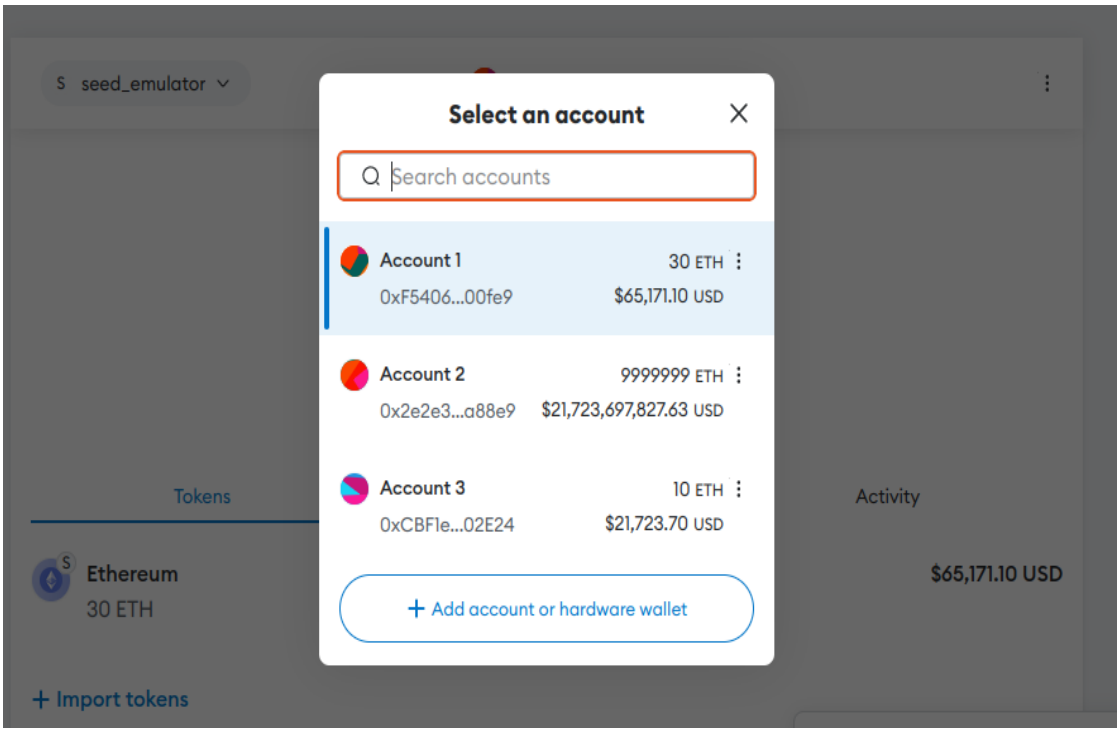




Task 1.c. Adding accounts

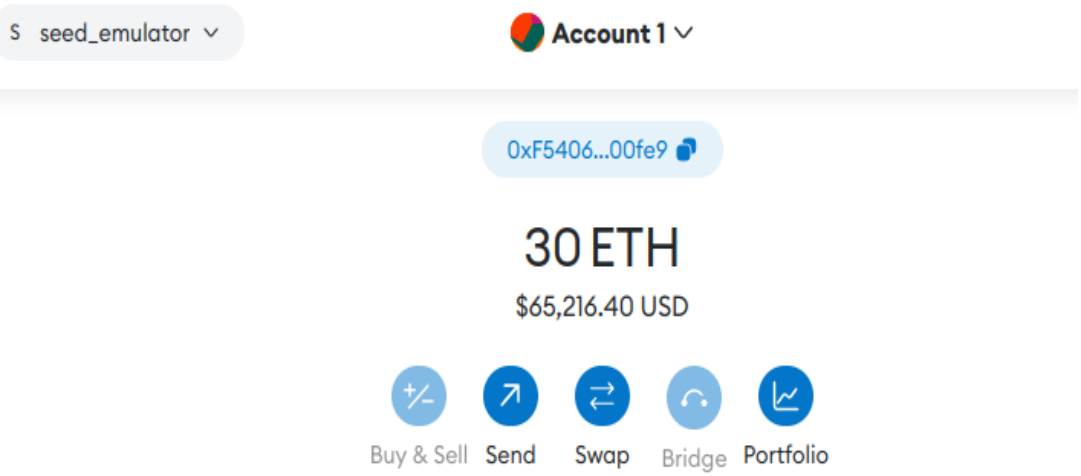
To add accounts, initially go to lock metamask, and try to reset the password using the mnemonics and when once successfully logged in, should be able to see 3 accounts with balance as shown below.

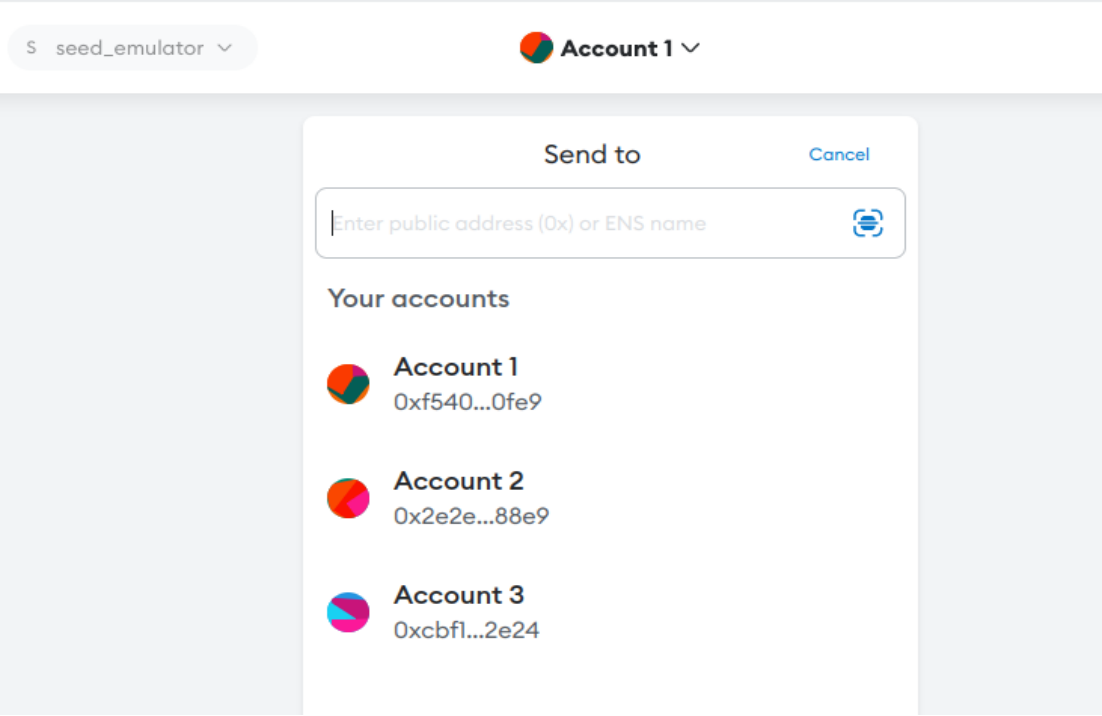




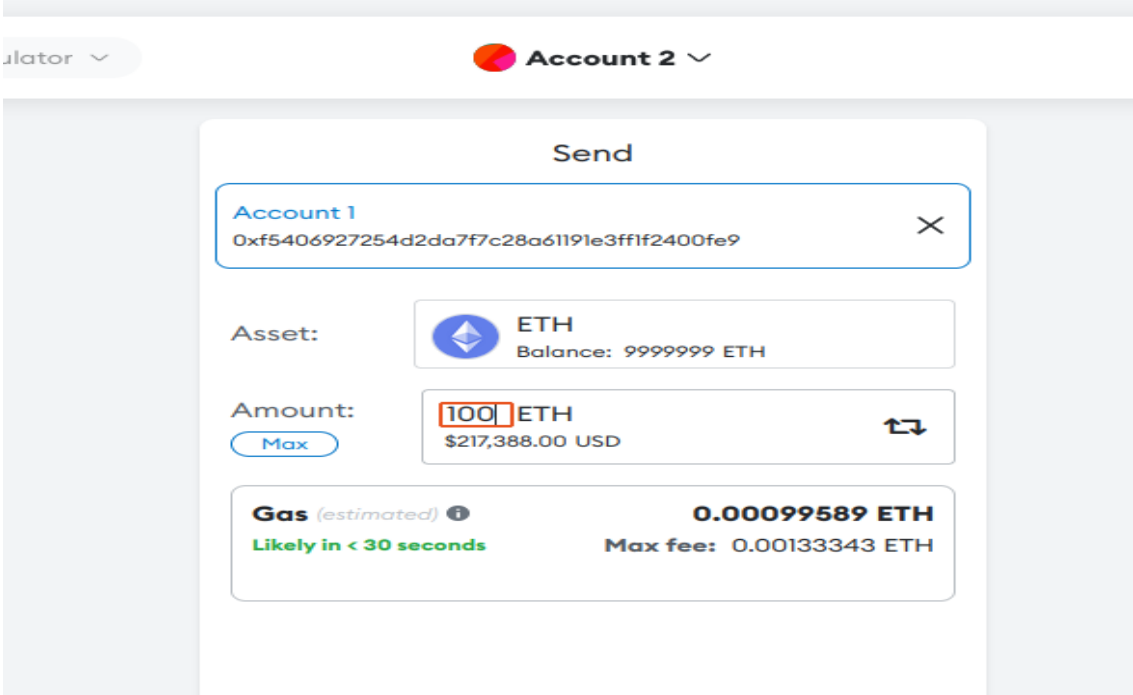
Task 1.d. Sending transactions

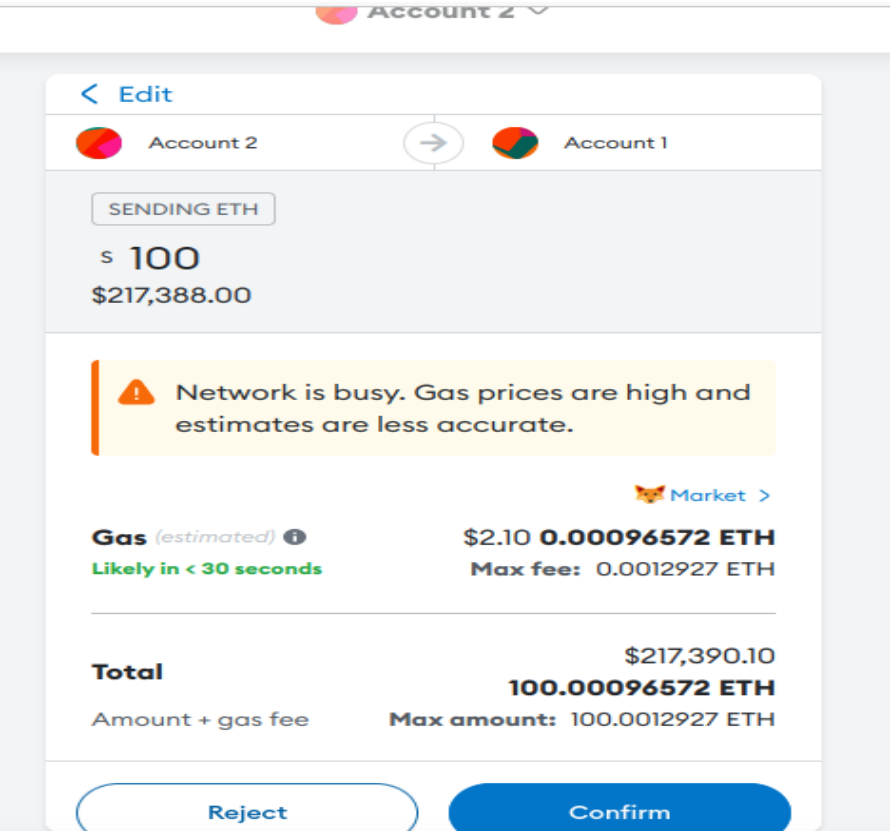
To send money to different accounts, click on send ad select the account you want to send money to.



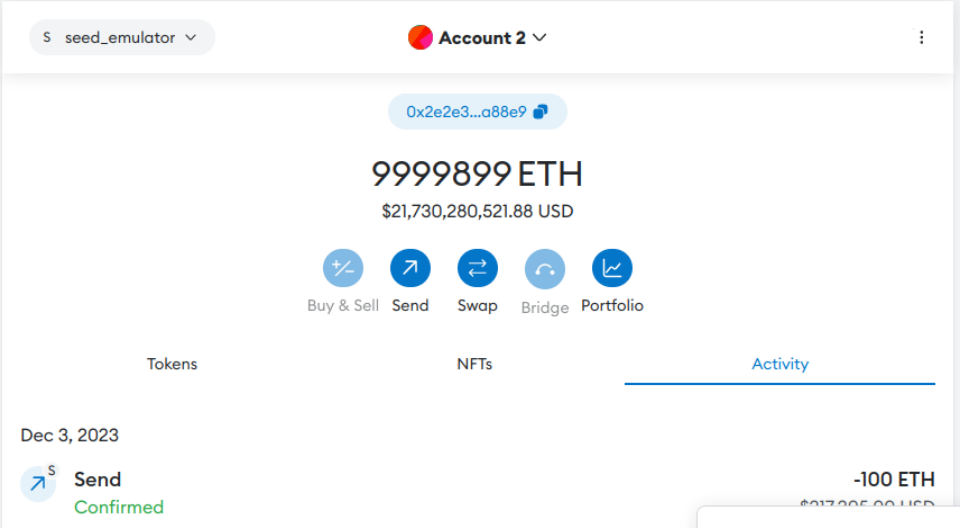


Enter the amount you would like to transact. And click on confirm.





Can see that the money is sent successfully.



Task 2: Interacting with Blockchain Using Python

Task 2.a: Installing Python modules

Install the web3 and docker modules.

```
[12/03/23]seed@VM:~/.../emulator_10$ pip3 install web3==5.31.1 docker
Collecting web3==5.31.1
```

```
Successfully installed aiohttp-3.9.1 aiosignal-1.3.1 async-timeout-4.0.3 attrs-2
3.1.0 base58-2.1.1 bitarray-2.8.3 cytoolz-0.12.2 docker-6.1.3 eth-abi-2.2.0 eth-
account-0.5.9 eth-hash-0.5.2 eth-keyfile-0.5.1 eth-keys-0.3.4 eth-rlp-0.2.1 eth-
typing-2.3.0 eth-utils-1.10.0 frozenlist-1.4.0 hexbytes-0.3.1 importlib-resource
s-6.1.1 ipfshttpclient-0.8.0a2 jsonschema-4.20.0 jsonschema-specifications-2023.
11.2 lru-dict-1.3.0 multiaddr-0.0.9 multidict-6.0.4 netaddr-0.9.0 packaging-23.2
 parsimonious-0.8.1 pkgutil-resolve-name-1.3.10 protobuf-3.19.5 referencing-0.31
.1 rlp-2.0.1 rpds-py-0.13.2 toolz-0.12.0 urllib3-2.1.0 varint-1.0.2 web3-5.31.1
websocket-client-1.7.0 websockets-9.1 yarl-1.9.3 zipp-3.17.0
[12/03/23]seed@VM:~/.../emulator_10$
```

Open web3_balance.py ~/Downloads/Labsetup/Files

```python
#!/bin/env python3
from web3 import Web3

url  = 'http://10.150.0.71:8545'
web3 = Web3(Web3.HTTPProvider(url))  # Connect to a blockchain node

addr = Web3.toChecksumAddress('0xF5406927254d2dA7F7c28A61191e3Ff1f2400fe9')
balance = web3.eth.get_balance(addr) # Get the balance
print(addr + ": " + str(Web3.fromWei(balance, 'ether')) + " ETH")
```

Task 2.b: Checking account balance

Open the web3.py file under the Files folder, and pass the account id for each one of them and then run the web3_balance.py file to know the balance of each account.

Open *web3_balance.py ~/Downloads/Labsetup/Files

```python
#!/bin/env python3
from web3 import Web3

url  = 'http://10.150.0.71:8545'
web3 = Web3(Web3.HTTPProvider(url))  # Connect to a blockchain node

addr = Web3.toChecksumAddress('0xF5406927254d2dA7F7c28A61191e3Ff1f2400fe9')
balance = web3.eth.get_balance(addr) # Get the balance
print(addr + ": " + str(Web3.fromWei(balance, 'ether')) + " ETH")
```

Open *web3_balance.py ~/Downloads/Labsetup/Files

```python
#!/bin/env python3
from web3 import Web3

url  = 'http://10.150.0.71:8545'
web3 = Web3(Web3.HTTPProvider(url))  # Connect to a blockchain node

addr = Web3.toChecksumAddress('0x2e2e3a61daC1A2056d9304F79C168cD16aAa88e9')
balance = web3.eth.get_balance(addr) # Get the balance
print(addr + ": " + str(Web3.fromWei(balance, 'ether')) + " ETH")
```

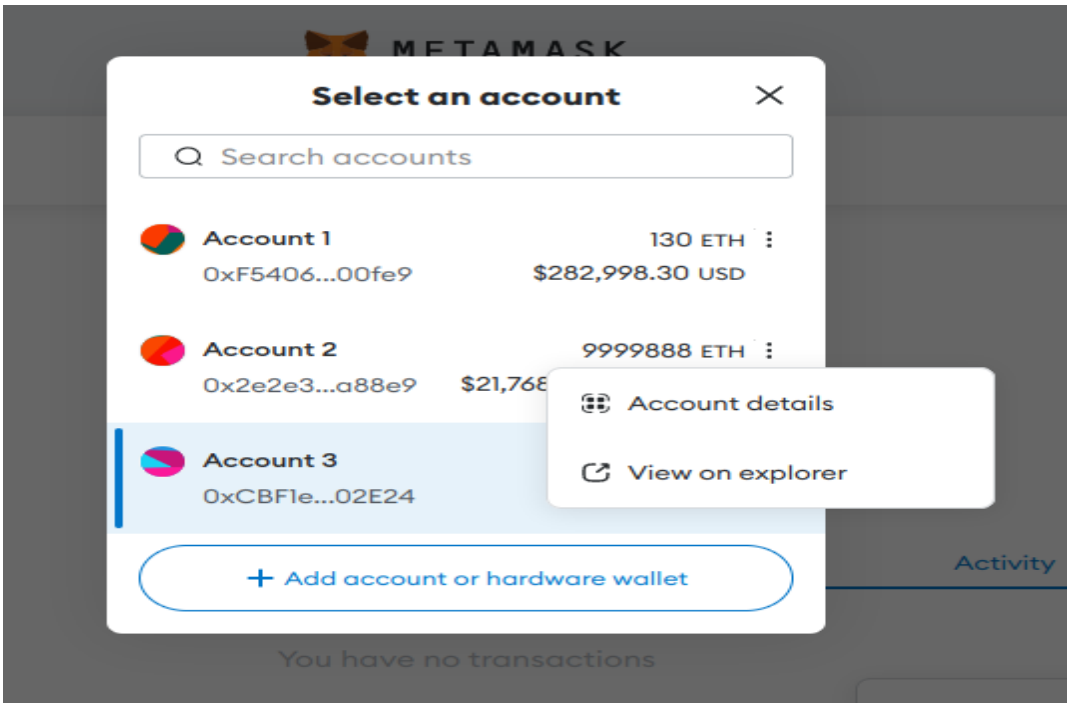
Open *web3_balance.py ~/Downloads/Labsetup/Files

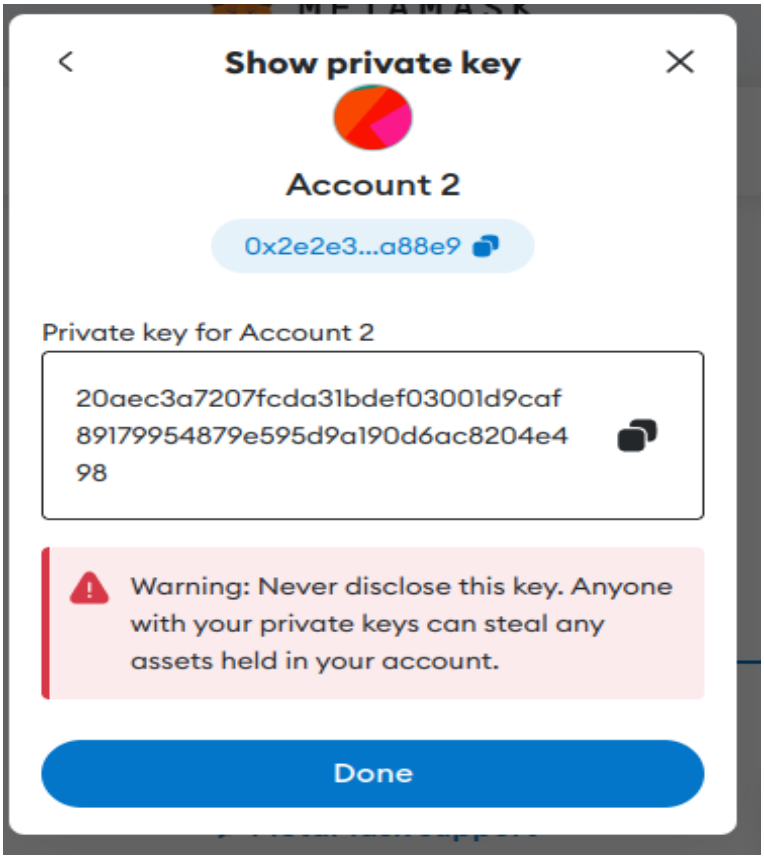
```python
#!/bin/env python3
from web3 import Web3

url  = 'http://10.150.0.71:8545'
web3 = Web3(Web3.HTTPProvider(url))  # Connect to a blockchain node

addr = Web3.toChecksumAddress('0xCBF1e330F0abD5c1ac979CF2B2B874cfD4902E24')
balance = web3.eth.get_balance(addr) # Get the balance
print(addr + ": " + str(Web3.fromWei(balance, 'ether')) + " ETH")
```

```
[12/03/23]seed@VM:~/.../Files$ python3 web3_balance.py
/usr/lib/python3/dist-packages/requests/__init__.py:89: RequestsDependencyWarning: urllib3 (2.1
.0) or chardet (3.0.4) doesn't match a supported version!
  warnings.warn("urllib3 ({}) or chardet ({}) doesn't match a supported "
0xF5406927254d2dA7F7c28A61191e3Ff1f2400fe9: 130 ETH
[12/03/23]seed@VM:~/.../Files$
[12/03/23]seed@VM:~/.../Files$
[12/03/23]seed@VM:~/.../Files$ python3 web3_balance.py
/usr/lib/python3/dist-packages/requests/__init__.py:89: RequestsDependencyWarning: urllib3 (2.1
.0) or chardet (3.0.4) doesn't match a supported version!
  warnings.warn("urllib3 ({}) or chardet ({}) doesn't match a supported "
0x2e2e3a61daC1A2056d9304F79C168cD16aAa88e9: 9999898.999968499999853 ETH
[12/03/23]seed@VM:~/.../Files$
[12/03/23]seed@VM:~/.../Files$ python3 web3_balance.py
/usr/lib/python3/dist-packages/requests/__init__.py:89: RequestsDependencyWarning: urllib3 (2.1
.0) or chardet (3.0.4) doesn't match a supported version!
  warnings.warn("urllib3 ({}) or chardet ({}) doesn't match a supported "
0xCBF1e330F0abD5c1ac979CF2B2B874cfD4902E24: 10 ETH
```

Task 2.c: Sending transactions

Navigate to account details and get the private key for an account.



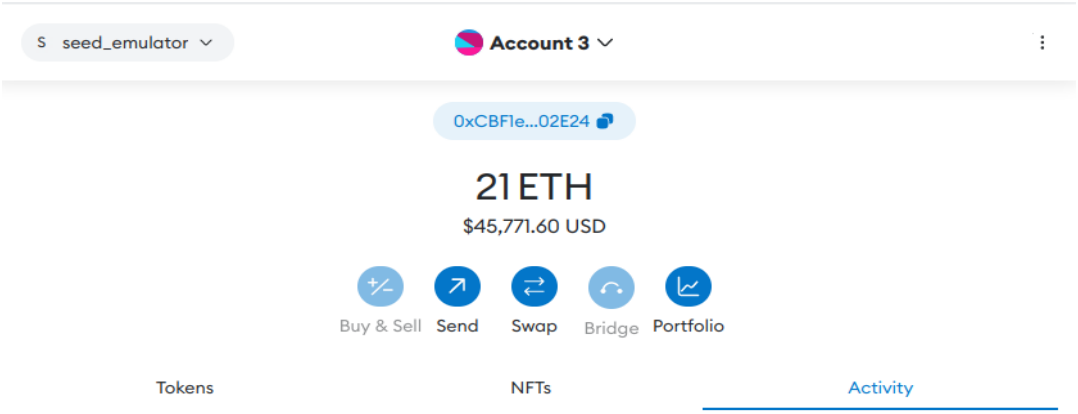


Add the private key, account id and the ip address in the web3_raw_tx.py file

web3_balance.py | web3_raw_tx.py

```python
#!/bin/env python3
from web3 import Web3
from eth_account import Account

web3 = Web3(Web3.HTTPProvider('http://10.162.0.71:8545'))

# Sender's private key
key = '20aec3a7207fcda31bdef03001d9caf89179954879e595d9a190d6ac8204e498'
sender = Account.from_key(key)

recipient = Web3.toChecksumAddress('0xCBF1e330F0abD5c1ac979CF2B2B874cfD4902E24')
tx = {
  'chainId':  1337,
  'nonce':    web3.eth.getTransactionCount(sender.address),
  'from':     sender.address,
  'to':       recipient,
  'value':    Web3.toWei("11", 'ether'),
  'gas':      200000,
  'maxFeePerGas':         Web3.toWei('4', 'gwei'),
  'maxPriorityFeePerGas': Web3.toWei('3', 'gwei'),
  'data':     ''
}
```

```
[12/03/23]seed@VM:~/.../Files$ python3 web3_raw_tx.py
/usr/lib/python3/dist-packages/requests/__init__.py:89: RequestsDependencyWarning: urllib3 (2.1
.0) or chardet (3.0.4) doesn't match a supported version!
  warnings.warn("urllib3 ({}) or chardet ({}) doesn't match a supported "
Transaction sent, waiting for receipt ...
Transaction Receipt: AttributeDict({'blockHash': HexBytes('0x1f26a30e7e1eb40ece4c7fe1fdc845a812
14d170800ff03a55289f979c6645a6'), 'blockNumber': 253, 'contractAddress': None, 'cumulativeGasUs
ed': 21000, 'effectiveGasPrice': 3000000007, 'from': '0x2e2e3a61daC1A2056d9304F79C168cD16aAa88e
9', 'gasUsed': 21000, 'logs': [], 'logsBloom': HexBytes('0x00000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0'), 'status': 1, 'to': '0xCBF1e330F0abD5c1ac979CF2B2B874cfD4902E24', 'transactionHash': HexByt
es('0xffc55952575ab0bce6b55d4c5f0990d8f6d1094179197b865b0156db2f87bd23'), 'transactionIndex': 0
, 'type': '0x2'})
```



Run the web3_raw_tx.py file and check the balance of the account. If successful, the transaction should be completed.

Task 3: Interacting with Blockchain Using Geth

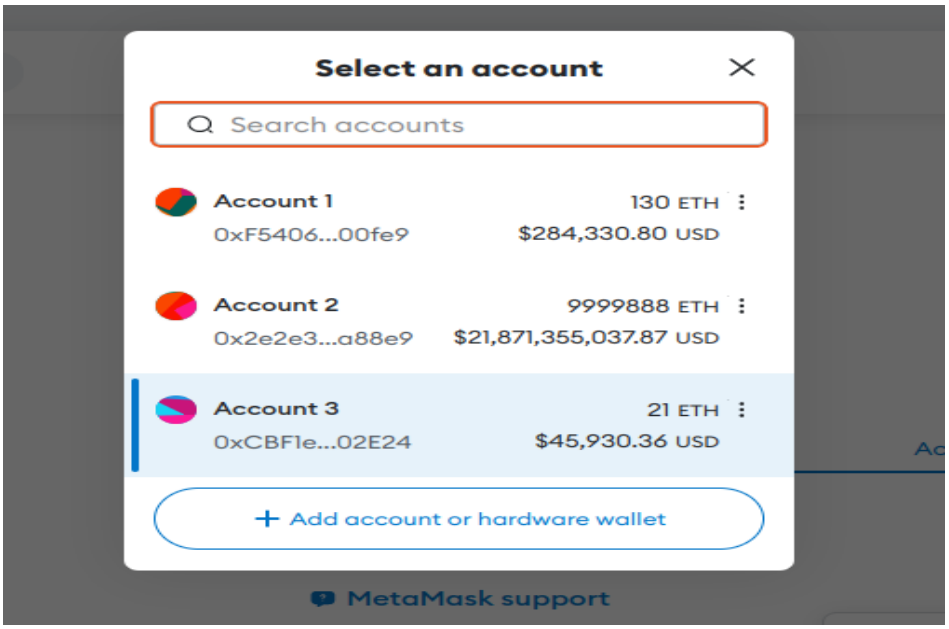
Login to a container, and use geth attach command to interact with the blockchain

```
[12/03/23]seed@VM:~/.../Labsetup$ docksh 7385
root@73858b6b42a7 / # geth attach
Welcome to the Geth JavaScript console!

instance: Geth/NODE_8/v1.10.26-stable-e5eb32ac/linux-amd64/go1.18.10
coinbase: 0xa888497f7938825f80f35867a1e707f42b9b347d
at block: 282 (Sun Dec 03 2023 21:59:44 GMT+0000 (UTC))
 datadir: /root/.ethereum
 modules: admin:1.0 clique:1.0 debug:1.0 engine:1.0 eth:1.0 miner:1.0 net:1.0 personal:1.0 rpc:
1.0 txpool:1.0 web3:1.0

To exit, press ctrl-d or type exit
```

Task 3.a: Getting balance

Get the balance of each account, using the myaccount API.

```
> myaccount = "0xF5406927254d2dA7F7c28A61191e3Ff1f2400fe9"
"0xF5406927254d2dA7F7c28A61191e3Ff1f2400fe9"
> eth.getBalance(myaccount)
130000000000000000000
> myaccount = "0x2e2e3a61daC1A2056d9304F79C168cD16aAa88e9"
"0x2e2e3a61daC1A2056d9304F79C168cD16aAa88e9"
> eth.getBalance(myaccount)
9.999887999905499999706e+24
> myaccount = "0xCBF1e330F0abD5c1ac979CF2B2B874cfD4902E24"
"0xCBF1e330F0abD5c1ac979CF2B2B874cfD4902E24"
> eth.getBalance(myaccount)
210000000000000000000
```



Task 3.b: Sending transactions

Using an account that is stored on the /root/ .ethereum/keystore, send money to an account on the metamask. We can see that the transaction was successful.

```
> sender = eth.accounts[0]
"0xa888497f7938825f80f35867a1e707f42b9b347d"
> target = "0xF5406927254d2dA7F7c28A61191e3Ff1f2400fe9"
"0xF5406927254d2dA7F7c28A61191e3Ff1f2400fe9"
> amount = web3.toWei(0.2, "ether")
"200000000000000000"
> eth.sendTransaction ({from: sender, to: target, value: amount})
"0x88ccabee849926b50ae892bc66c4e84aed9cdedeb2b749178818da8d475fb55c"
```

Task 3.c: Sending transactions from a different account

Use an account from metamask to send transactions to different accounts. And can see that transaction failed and cannot send money from a metamask account.

```
> sender = "0x2e2e3a61daC1A2056d9304F79C168cD16aAa88e9"
"0x2e2e3a61daC1A2056d9304F79C168cD16aAa88e9"
> target = "0xF5406927254d2dA7F7c28A61191e3Ff1f2400fe9"
"0xF5406927254d2dA7F7c28A61191e3Ff1f2400fe9"
>  amount = web3.toWei(0.2, "ether")
"200000000000000000"
> eth.sendTransaction ({from: sender, to: target, value: amount})
Error: unknown account
        at web3.js:6365:9(45)
        at send (web3.js:5099:62(34))
        at <eval>:1:21(10)
```

Task 4: Adding a Full Node

Login to the new node and navigate to the tmp folder.

```
[12/03/23]seed@VM:~/.../emulator_10$ dockps | grep new
3daec960e76b  as150h-new_eth_node-10.150.0.74
[12/03/23]seed@VM:~/.../emulator_10$ docksh 3da
root@3daec960e76b / # cd tmp
root@3daec960e76b /tmp # touch eth-genesis.json
root@3daec960e76b /tmp # nano eth-genesis.json
root@3daec960e76b /tmp # cat eth-genesis.json
{"config": {"chainId": 1337, "homesteadBlock": 0, "eip150Block": 0, "eip150Hash": "0x0000000000000000000000
0000000000000000000000000000000000000000", "eip155Block": 0, "eip158Block": 0, "byzantiumBlock": 0, "cons
tantinopleBlock": 0, "petersburgBlock": 0, "istanbulBlock": 0, "muirGlacierBlock": 0, "berlinBlock": 0, "lo
ndonBlock": 0, "arrowGlacierBlock": 0, "grayGlacierBlock": 0, "clique": {"period": 15, "epoch": 30000}}, "n
once": "0x0", "timestamp": "0x655432eb", "extraData": "0x0000000000000000000000000000000000000000000000000
000000000000001be9288F9a7D2F809250f15c487E3a5A9Cf71f4F8c400205fDb103431F6aC7409655ad3cf8f6d00772943017A1fa5
f255fC0f06625Aec22319FCd5b35449ba5C5F185e9694146d60cFE72681E2158499A38a1FA3aF6779A79268e652027120E8De51f6A4
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000", "gasLimit": "0x47b760", "difficulty": "0x1", "mixHash": "0x0000000000000000000000
0000000000000000000000000000000000000000", "coinbase": "0x0000000000000000000000000000000000000000", "all
oc": {"F5406927254d2dA7F7c28A61191e3Ff1f2400fe9": {"balance": "3000000000000000000"}, "2e2e3a61daC1A2056d9
304F79C168cD16aAa88e9": {"balance": "99999999000000000000000"}, "CBF1e330F0abD5c1ac979CF2B2B874cfD4902E24
": {"balance": "1000000000000000000"}, "A2a28c011e281CA0dA0D878A82d854FD789C154c": {"balance": "1000000000
00000000000"}, "513C434dBA61AE5CFEf4552daC2D2f85450870aA": {"balance": "10000000000000000000"}, "BaED4A4Ff
fff4e047B8a39F00284732eF6244f4B": {"balance": "1000000000000000000"}, "F5D434D36dD2bF53d2D1dB4FD40076A0C1
C44F8d": {"balance": "10000000000000000000"}, "3eE934c34747460C7083Ad052bab58E6DF5dbd84": {"balance": "100
00000000000000000"}, "94Bfb5a96B191011892e23bceB2d1ae22B4f1C25": {"balance": "10000000000000000000"}, "46
8DaF3c6E9a79255F4b2985A3801C791Aa9037d": {"balance": "10000000000000000000"}, "f77d3Bb88460C58784c3112A128
9D68105e28f60": {"balance": "10000000000000000000"}, "477a4e1fcdF12Cb8bAba4eAD4c43F1fF26cCeD12": {"balance
": "10000000000000000000"}, "830EB42863505ACf1127905C835B6A7a36760Fa0": {"balance": "10000000000000000000
"}, "1be9288F9a7D2F809250f15c487E3a5A9Cf71f4F": {"balance": "32000000000000000000"}, "1081c645CC8c21EfbB011
4eAc5fcDBE01a1a4b19": {"balance": "32000000000000000000"}, "8c400205fDb103431F6aC7409655ad3cf8f6d007": {"ba
lance": "32000000000000000000"}, "D4CC43e3f2830f9082495Dba904B57fc2Ca95CBd": {"balance": "32000000000000000
000"}, "72943017A1fa5f255fC0f06625Aec22319FCd5b3": {"balance": "32000000000000000000"}, "C5247277519ca71C48
8e7D093350aa659aCaDF7e": {"balance": "32000000000000000000"}, "5449ba5C5F185e9694146d60cFE72681E2158499": {
"balance": "32000000000000000000"}, "a888497f7938825F80F35867a1E707F42b9b347D": {"balance": "32000000000000
000000"}, "A38a1FA3aF6779A79268e652027120E8De51f6A4": {"balance": "32000000000000000000"}, "8e9Be21Be75EF64
```

Navigate to one of the eth container, and under the tmp folder copy the contents of the file eth-genesis.json file

```
[12/03/23]seed@VM:~/.../emulator_10$ dockps | grep Eth
73858b6b42a7  as162h-Ethereum-POA-07-10.162.0.71
fd0fd716801c  as160h-Ethereum-POA-05-10.160.0.71
d4e7ec3b4823  as161h-Ethereum-POA-06-Signer-BootNode-10.161.0.71
430856ee26af  as152h-Ethereum-POA-02-Signer-10.152.0.71
7ba6e2119aad  as163h-Ethereum-POA-08-Signer-10.163.0.71
bb6f910223be  as164h-Ethereum-POA-09-BootNode-10.164.0.71
d2429c0bef63  as151h-Ethereum-POA-01-10.151.0.71
342a7352cbdb  as150h-Ethereum-POA-00-Signer-BootNode-10.150.0.71
cdabb32b7a9a  as154h-Ethereum-POA-04-Signer-10.154.0.71
10b863a34d80  as153h-Ethereum-POA-03-BootNode-10.153.0.71
[12/03/23]seed@VM:~/.../emulator_10$ docksh fd0
root@fd0fd716801c / # cd tmp
root@fd0fd716801c /tmp # ls
eth-bootstrapper  eth-nodes      eth-password
eth-genesis.json  eth-node-urls  keystore
root@fd0fd716801c /tmp # cat eth-genesis.json
{"config": {"chainId": 1337, "homesteadBlock": 0, "eip150Block": 0, "eip150Hash": "0x0000000000000000000000000000000000000000000000000
000000000000000", "eip155Block": 0, "eip158Block": 0, "byzantiumBlock": 0, "constantinopleBlock": 0, "petersburgBlock": 0, "istanbulBl
ock": 0, "muirGlacierBlock": 0, "berlinBlock": 0, "londonBlock": 0, "arrowGlacierBlock": 0, "grayGlacierBlock": 0, "clique": {"period"
: 15, "epoch": 30000}}, "nonce": "0x0", "timestamp": "0x655432eb", "extraData": "0x0000000000000000000000000000000000000000000000000
000000000000001be9288F9a7D2F809250f15c487E3a5A9Cf71f4F8c400205fDb103431F6aC7409655ad3cf8f6d00772943017A1fa5f255fC0f06625Aec22319FCd5b35
449ba5C5F185e9694146d60cFE72681E2158499A38a1FA3aF6779A79268e652027120E8De51f6A4000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
"0x0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000", "coinbase": "0x0000000000000000000000000000000000000000", "alloc
": {"F5406927254d2dA7F7c28A61191e3Ff1f2400fe9": {"balance": "3000000000000000000"}, "2e2e3a61daC1A2056d9304F79C168cD16aAa88e9": {"bal
ance": "99999999000000000000000"}, "CBF1e330F0abD5c1ac979CF2B2B874cfD4902E24": {"balance": "1000000000000000000"}, "A2a28c011e281CA
0dA0D878A82d854FD789C154c": {"balance": "10000000000000000000"}, "513C434dBA61AE5CFEf4552daC2D2f85450870aA": {"balance": "10000000000
0000000000"}, "BaED4A4Ffffff4e047B8a39F00284732eF6244f4B": {"balance": "10000000000000000000"}, "F5D434D36dD2bF53d2D1dB4FD40076A0C1C44
F8d": {"balance": "10000000000000000000"}, "3eE934c34747460C7083Ad052bab58E6DF5dbd84": {"balance": "10000000000000000000"}, "94Bfb5a
96B191011892e23bceB2d1ae22B4f1C25": {"balance": "10000000000000000000"}, "468DaF3c6E9a79255F4b2985A3801C791Aa9037d": {"balance": "100
00000000000000000"}, "f77d3Bb88460C58784c3112A1289D68105e28f60": {"balance": "10000000000000000000"}, "477a4e1fcdF12Cb8bAba4eAD4c43F
1fF26cCeD12": {"balance": "10000000000000000000"}, "830EB42863505ACf1127905C835B6A7a36760Fa0": {"balance": "10000000000000000000"},
"1be9288F9a7D2F809250f15c487E3a5A9Cf71f4F": {"balance": "32000000000000000000"}, "1081c645CC8c21EfbB0114eAc5fcDBE01a1a4b19": {"balance
": "32000000000000000000"}, "8c400205fDb103431F6aC7409655ad3cf8f6d007": {"balance": "32000000000000000000"}, "D4CC43e3f2830f9082495Dba
904B57fc2Ca95CBd": {"balance": "32000000000000000000"}, "72943017A1fa5f255fC0f06625Aec22319FCd5b3": {"balance": "32000000000000000000"
}, "C5247277519ca71C488e7D093350aa659aCaDF7e": {"balance": "32000000000000000000"}, "5449ba5C5F185e9694146d60cFE72681E2158499": {"bala
nce": "32000000000000000000"}, "a888497f7938825F80F35867a1E707F42b9b347D": {"balance": "32000000000000000000"}, "A38a1FA3aF6779A79268e
652027120E8De51f6A4": {"balance": "32000000000000000000"}, "8e9Be21Be75EF64067F8fBFc0379a687eE65b867": {"balance": "32000000000000000000
00"}}, "number": "0x0", "gasUsed": "0x0", "parentHash": "0x0000000000000000000000000000000000000000000000000000000000000000", "baseFee
PerGas": null}
```

Create a file in the new node tmp folder with the same name eth-genesis.json and paste the contents of the file copied earlier. Using the command shown below execute the file

```
root@3daec960e76b /tmp # geth --datadir /root/.ethereum init /eth-genesis.json
Fatal: Failed to read genesis file: open /eth-genesis.json: no such file or directory
1 root@3daec960e76b /tmp # geth --datadir /root/.ethereum init eth-genesis.json
INFO [12-03|22:47:19.014] Maximum peer count                       ETH=50 LES=0 total=50
INFO [12-03|22:47:19.044] Smartcard socket not found, disabling    err="stat /run/pcscd/pcscd.comm: no such
 file or directory"
INFO [12-03|22:47:19.093] Set global gas cap                       cap=50,000,000
INFO [12-03|22:47:19.108] Allocated cache and file handles         database=/root/.ethereum/geth/chaindata
cache=16.00MiB handles=16
INFO [12-03|22:47:20.075] Opened ancient database                  database=/root/.ethereum/geth/chaindata/
ancient/chain readonly=false
INFO [12-03|22:47:20.076] Writing custom genesis block
INFO [12-03|22:47:20.105] Persisted trie from memory database      nodes=33 size=4.78KiB time=9.151049ms gc
nodes=0 gcsize=0.00B gctime=0s livenodes=1 livesize=0.00B
INFO [12-03|22:47:20.107] Successfully wrote genesis state         database=chaindata
hash=1bfc81..11048e
INFO [12-03|22:47:20.110] Allocated cache and file handles         database=/root/.ethereum/geth/lightchain
data cache=16.00MiB handles=16
INFO [12-03|22:47:20.751] Opened ancient database                  database=/root/.ethereum/geth/lightchain
data/ancient/chain readonly=false
INFO [12-03|22:47:20.751] Writing custom genesis block
INFO [12-03|22:47:21.032] Persisted trie from memory database      nodes=33 size=4.78KiB time=213.990851ms
gcnodes=0 gcsize=0.00B gctime=0s livenodes=1 livesize=0.00B
INFO [12-03|22:47:21.041] Successfully wrote genesis state         database=lightchaindata
     hash=1bfc81..11048e
```

b. Copy the contents of the file eth-node-urls from the node container.

```
root@fd0fd716801c /tmp # cat eth-node-urls
enode://ff4cb85054a0d940d33ccea348cda5c87ba8ed1e03ab328a76dc9b0e78f39c57012151fdb656cb5e300927c2684ae9856542ca06a4294e8c8ce879da344da2b3@10.150.0
.71:30301,
enode://7e495dc92b061bb3b5eb35248938d2eb99c79ec548bee8a0a85ad50957a3c9e1b5d633082d59e2b99aab946636882dc033801ffa2e2dfcecadd1c1272edf5b67@10.153.0
.71:30301,
enode://de9e0eaca6f47d4ee2ca9cbd3e631bec72db6cdf18062d6f45519f523161a562a24a61e1c543ee5dfdbae1d628a7f7189aa7d869a9dca666347b010c911f10e6@10.161.0
.71:30301,
enode://b498faf9206d57a88e9a8a9f37483da9e69681e72ed8b8e994d5d2a616ec015fd11497871d07969f94265add2adbf6fe2ee7fb7f4e6b1e197ddbfff2f1e83868@10.164.0
.71:30301,
enode://ff4cb85054a0d940d33ccea348cda5c87ba8ed1e03ab328a76dc9b0e78f39c57012151fdb656cb5e300927c2684ae9856542ca06a4294e8c8ce879da344da2b3@10.150.0
.71:30301,
enode://7e495dc92b061bb3b5eb35248938d2eb99c79ec548bee8a0a85ad50957a3c9e1b5d633082d59e2b99aab946636882dc033801ffa2e2dfcecadd1c1272edf5b67@10.153.0
.71:30301,
enode://de9e0eaca6f47d4ee2ca9cbd3e631bec72db6cdf18062d6f45519f523161a562a24a61e1c543ee5dfdbae1d628a7f7189aa7d869a9dca666347b010c911f10e6@10.161.0
.71:30301,
enode://b498faf9206d57a88e9a8a9f37483da9e69681e72ed8b8e994d5d2a616ec015fd11497871d07969f94265add2adbf6fe2ee7fb7f4e6b1e197ddbfff2f1e83868@10.164.0
.71:30301,
root@fd0fd716801c /tmp #
```

Create a file under the tmp folder with the same name eth-node-urls in the full node container.

Execute the file using the command shown below.

```
root@3daec960e76b /tmp # touch eth-node-urls
root@3daec960e76b /tmp # nano eth-node-urls
root@3daec960e76b /tmp #
root@3daec960e76b /tmp # nano eth-node-urls
root@3daec960e76b /tmp # geth --datadir /root/.ethereum --identity="NEW_NODE_01" --networkid=1337 \
--syncmode full --snapshot=False --verbosity=2 --port 30303 \
--bootnodes "$(cat /tmp/eth-node-urls)" --allow-insecure-unlock \
--http --http.addr 0.0.0.0 --http.corsdomain "*" \
--http.api web3,eth,debug,personal,net,clique,engine,admin,txpool
WARN [12-03|22:52:00.932] Error reading unclean shutdown markers   error="leveldb: not found"
WARN [12-03|22:52:00.933] Engine API enabled                       protocol=eth
WARN [12-03|22:52:00.935] Engine API started but chain not configured for merge yet
WARN [12-03|22:53:18.495] Served eth_coinbase                      reqid=3 duration="115.022µs" err="etherbase must be explicitly specified"


WARN [12-03|22:54:25.463] Please backup your key file!             path=/root/.ethereum/keystore/UTC--2023-12-03T22-53-49.537316787Z--99ece95e12f
833a0b1ded17f6f9dac563520baf3
WARN [12-03|22:54:25.464] Please remember your password!
```

```
[12/03/23]seed@VM:~/.../emulator_10$ dockps | grep new
3daec960e76b  as150h-new_eth_node-10.150.0.74
[12/03/23]seed@VM:~/.../emulator_10$ docksh 3da
root@3daec960e76b / # geth attach
Welcome to the Geth JavaScript console!

instance: Geth/NEW_NODE_01/v1.10.26-stable-e5eb32ac/linux-amd64/go1.18.10
at block: 467 (Sun Dec 03 2023 22:53:10 GMT+0000 (UTC))
 datadir: /root/.ethereum
 modules: admin:1.0 clique:1.0 debug:1.0 engine:1.0 eth:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 txpoo
l:1.0 web3:1.0

To exit, press ctrl-d or type exit
> personal.newAccount()
Passphrase:
Repeat passphrase:


"0x99ece95e12f833a0b1ded17f6f9dac563520baf3"
```

From the new full node run the geth attach command to connect to the blockchain. Use personal.newAccount() to create a new account on this node.

Modify the web3_raw_tx.py file so that it connects to the new node and can be used to make transactions. Add the recipient account id to send money from.

```
Open   web3_raw_tx.py   Save
~/Downloads/LabSetup/Files

#!/bin/env python3
from web3 import Web3
from eth_account import Account

web3 = Web3(Web3.HTTPProvider('http://10.162.0.71:8545'))

# Sender's private key
key = '20aec3a7207fcda31bdef03001d9caf89179954879e595d9a190d6ac8204e498'
sender = Account.from_key(key)

recipient = Web3.toChecksumAddress('0x99ece95e12f833a0b1ded17f6f9dac563520baf3')
tx = {
  'chainId':  1337,
  'nonce':    web3.eth.getTransactionCount(sender.address),
  'from':     sender.address,
  'to':       recipient,
  'value':    Web3.toWei("11", 'ether'),
  'gas':      200000,
  'maxFeePerGas':         Web3.toWei('4', 'gwei'),
  'maxPriorityFeePerGas': Web3.toWei('3', 'gwei'),
  'data':     ''
}

# Sign the transaction and send it out
signed_tx = web3.eth.account.sign_transaction(tx, sender.key)
```

Execute the python code and can be seen that it is successful.

```
seed@VM: ~/.../emulator_10   root@3daec960e76b: geth...   root@fd0fd716801c: /tmp   root@3daec960e76b: geth...   seed@VM: ~/.../Files
[12/03/23]seed@VM:~/.../emulator_10$ cd ..
[12/03/23]seed@VM:~/.../Labsetup$ cd Files/
[12/03/23]seed@VM:~/.../Files$ python3 web3_raw_tx.py
/usr/lib/python3/dist-packages/requests/__init__.py:89: RequestsDependencyWarning: urllib3 (2.1.0) or ch
ardet (3.0.4) doesn't match a supported version!
  warnings.warn("urllib3 ({}) or chardet ({}) doesn't match a supported "
Transaction sent, waiting for receipt ...
Transaction Receipt: AttributeDict({'blockHash': HexBytes('0xa3914ec10405dddd751f05a5425f9f1eabfb07452d2
639ff77763ead8fe11941'), 'blockNumber': 484, 'contractAddress': None, 'cumulativeGasUsed': 21000, 'effec
tiveGasPrice': 3000000007, 'from': '0x2e2e3a61daC1A2056d9304F79C168cD16aAa88e9', 'gasUsed': 21000, 'logs
': [], 'logsBloom': HexBytes('0x00000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000'), 'status': 1, 'to': '0x99eCe95E12F833a0B1dEd17f6f9DaC563520bAF3', 'transactio
nHash': HexBytes('0x2cdaaf7ca447cad3facee2ce7ce59c0ad2ef06a1b572e5cec041814ebc939130'), 'transactionInde
x': 0, 'type': '0x2'})
[12/03/23]seed@VM:~/.../Files$
```

Login to localhost:5000 to see the transaction on the blocks, and you can see it was successful.