Introduction to Blockchains

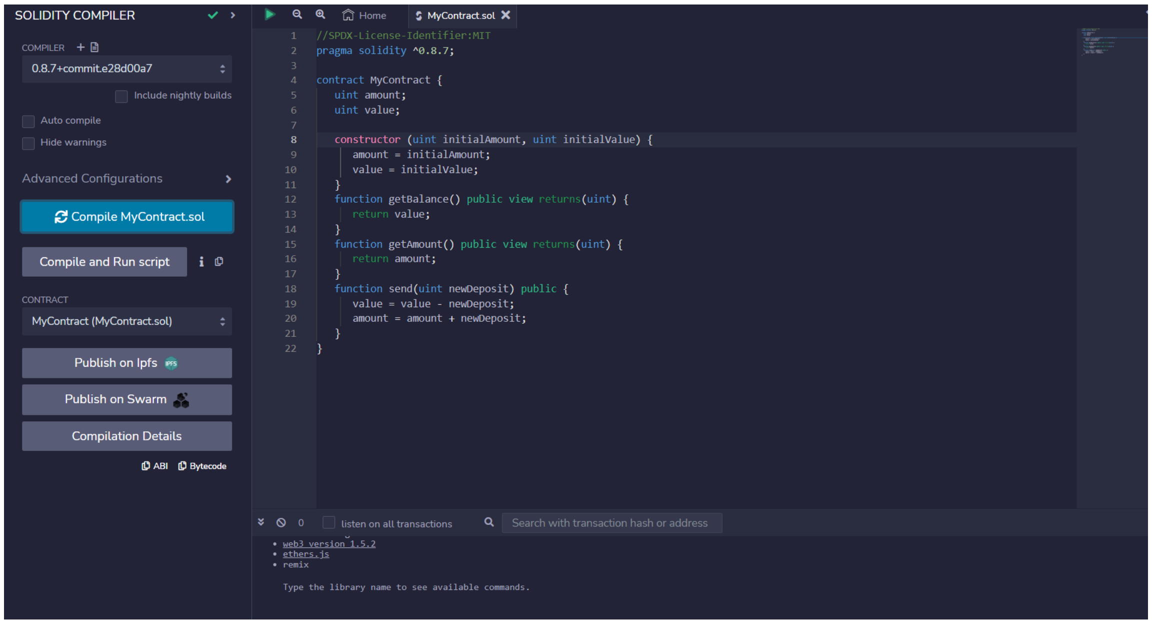
Assignment 1

Devdatt N(200010012)

Step 1:

1.1:

The given solidity code has been successfully compiled.



COMPILER: 0.8.7+commit.e28d00a7

CONTRACT: MyContract (MyContract.sol)

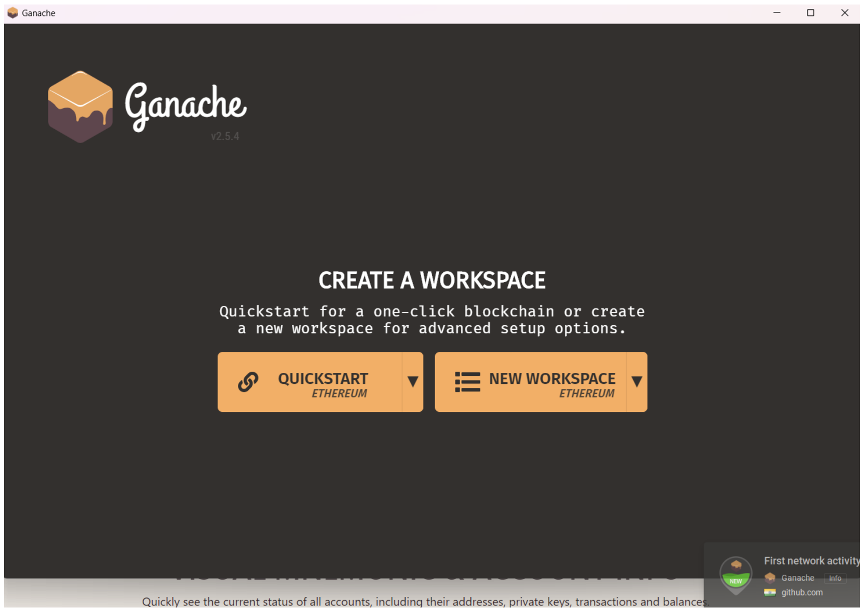
```
//SPDX-License-Identifier:MIT
pragma solidity ^0.8.7;

contract MyContract {
   uint amount;
   uint value;

   constructor (uint initialAmount, uint initialValue) {
      amount = initialAmount;
      value = initialValue;
   }
   function getBalance() public view returns(uint) {
      return value;
   }
   function getAmount() public view returns(uint) {
      return amount;
   }
   function send(uint newDeposit) public {
      value = value - newDeposit;
      amount = amount + newDeposit;
   }
}
```

```
• web3 version 1.5.2
• ethers.js
• remix

  Type the library name to see available commands.
```

Slight changes have been made to the code to support the 0.8.7 Solidity compiler.

1.2:

After downloading and installing Ganache from the website, the following screen appears:

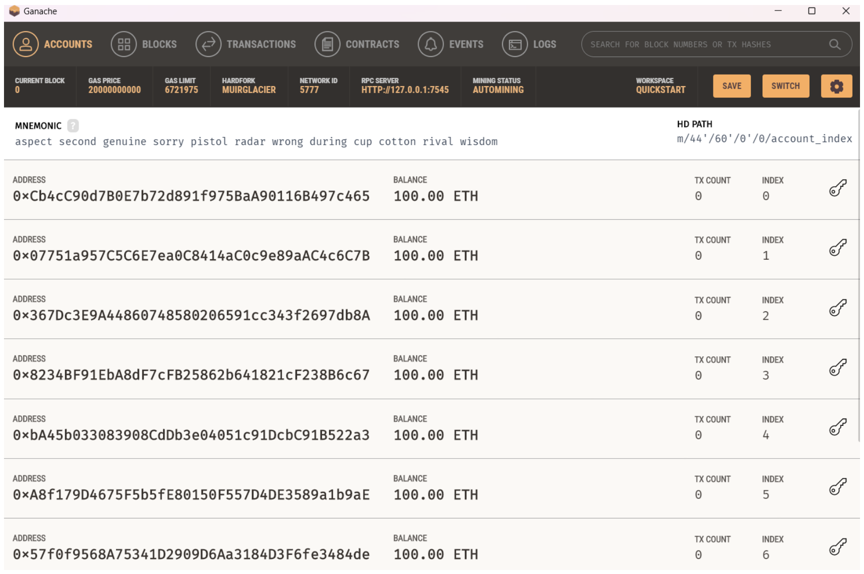


CREATE A WORKSPACE

Quickstart for a one-click blockchain or create a new workspace for advanced setup options.

Quickly see the current status of all accounts, including their addresses, private keys, transactions and balances

After clicking on Quickstart to start a locally hosted Ethereum network, we get the following screen:



CURRENT BLOCK	GAS PRICE	GAS LIMIT	HARDFORK	NETWORK ID	RPC SERVER	MINING STATUS	WORKSPACE
0	20000000000	6721975	MUIRGLACIER	5777	HTTP://127.0.0.1:7545	AUTOMINING	QUICKSTART

MNEMONIC
aspect second genuine sorry pistol radar wrong during cup cotton rival wisdom

HD PATH
m/44'/60'/0'/0/account_index

ADDRESS	BALANCE	TX COUNT	INDEX
0×Cb4cC90d7B0E7b72d891f975BaA90116B497c465	100.00 ETH	0	0
0×07751a957C5C6E7ea0C8414aC0c9e89aAC4c6C7B	100.00 ETH	0	1
0×367Dc3E9A448607485802065 91cc343f2697db8A	100.00 ETH	0	2
0×8234BF91EbA8dF7cFB25862b641821cF238B6c67	100.00 ETH	0	3
0×bA45b033083908CdDb3e04051c91DcbC91B522a3	100.00 ETH	0	4
0×A8f179D4675F5b5fE80150F557D4DE3589a1b9aE	100.00 ETH	0	5
0×57f0f9568A75341D2909D6Aa3184D3F6fe3484de	100.00 ETH	0	6

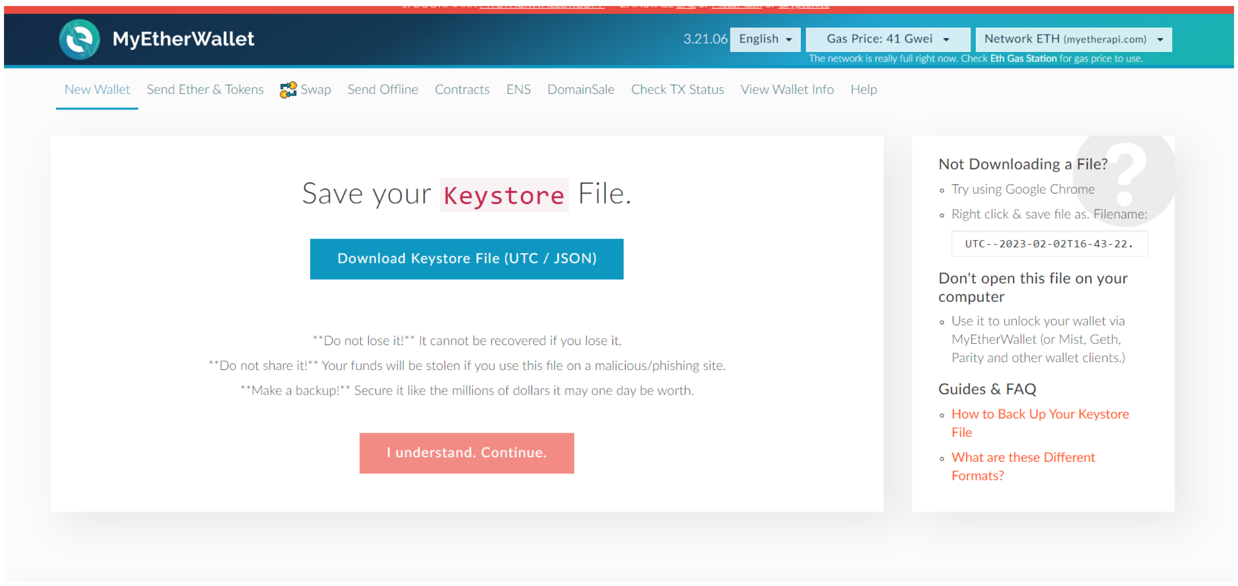
Step 2:

2.1:

MyEtherWallet has been downloaded from the link and has been extracted.

2.2:

After setting the password, the Ethereum wallet private key is created and can be downloaded. The following screen appears at the 'Download Keystore file' screen:



MyEtherWallet 3.21.06 English Gas Price: 41 Gwei Network ETH (myetherapi.com)

The network is really full right now. Check **Eth Gas Station** for gas price to use.

New Wallet Send Ether & Tokens Swap Send Offline Contracts ENS DomainSale Check TX Status View Wallet Info Help

Save your Keystore File.

Download Keystore File (UTC / JSON)

Do not lose it! It cannot be recovered if you lose it.

Do not share it! Your funds will be stolen if you use this file on a malicious/phishing site.

Make a backup! Secure it like the millions of dollars it may one day be worth.

I understand. Continue.

Not Downloading a File?

- Try using Google Chrome
- Right click & save file as. Filename:

```
UTC--2023-02-02T16-43-22.
```

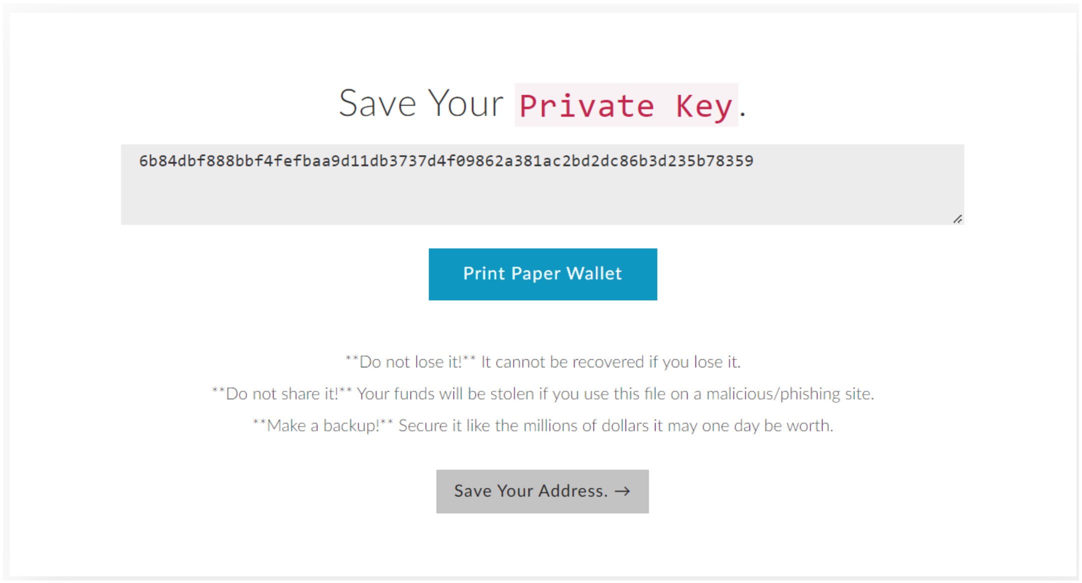
Don't open this file on your computer

- Use it to unlock your wallet via MyEtherWallet (or Mist, Geth, Parity and other wallet clients.)

Guides & FAQ

- How to Back Up Your Keystore File
- What are these Different Formats?

After downloading the keystore file, the following screen appears:



Save Your Private Key.

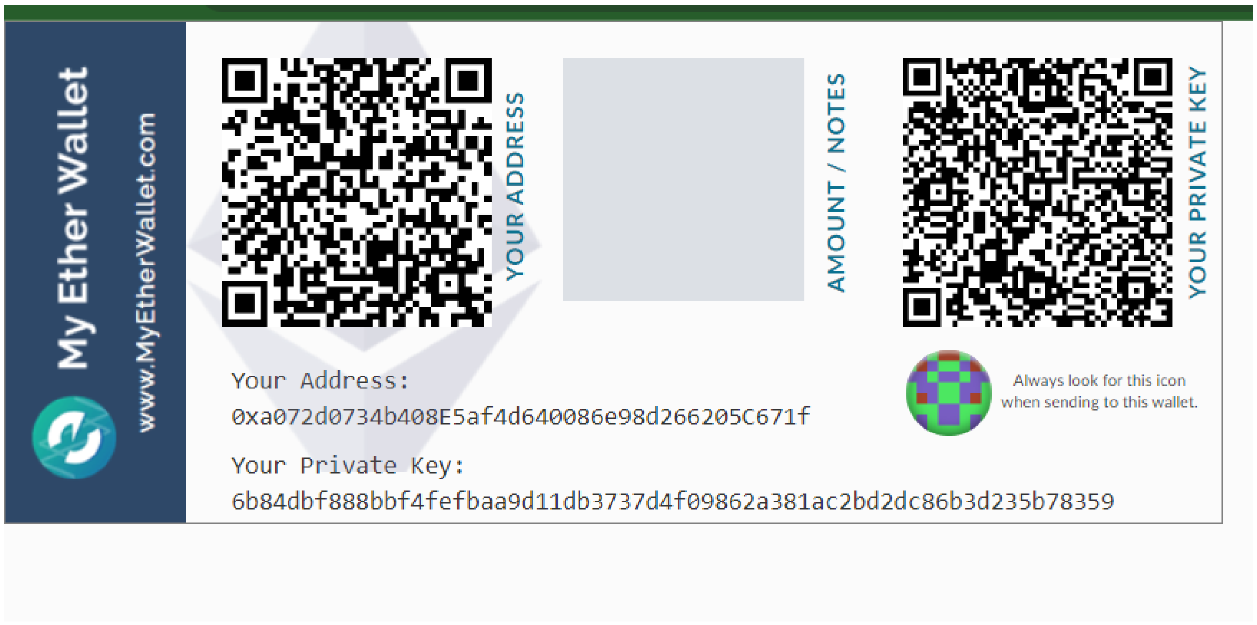
```
6b84dbf888bbf4fefbaa9d11db3737d4f09862a381ac2bd2dc86b3d235b78359
```

Do not lose it! It cannot be recovered if you lose it.

Do not share it! Your funds will be stolen if you use this file on a malicious/phishing site.

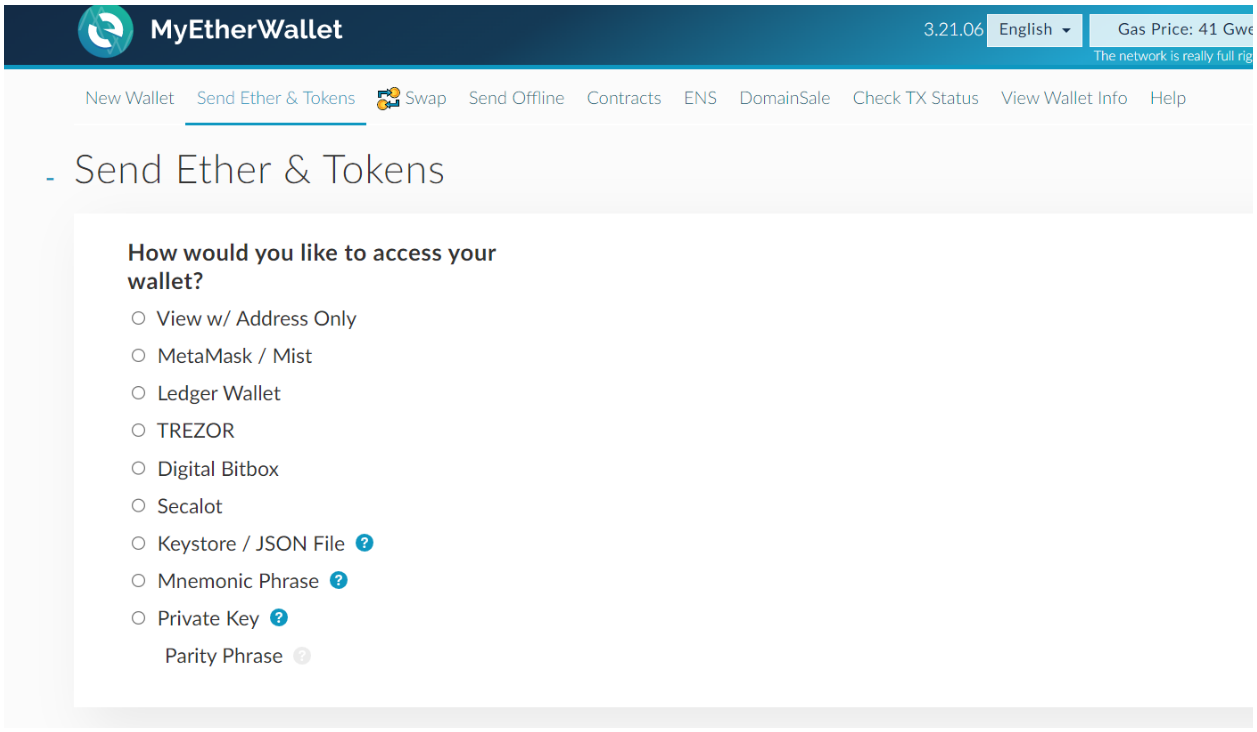
Make a backup! Secure it like the millions of dollars it may one day be worth.

Clicking on the 'Print Paper Wallet' option, I've also saved a PDF of the following screen:



```
Your Address:
0xa072d0734b408E5af4d640086e98d266205C671f
Your Private Key:
6b84dbf888bbf4fefbaa9d11db3737d4f09862a381ac2bd2dc86b3d235b78359
```

Now, on the ‘Send Ether & Tokens’ tab, we can see the ‘How would you like to access your wallet’ screen:

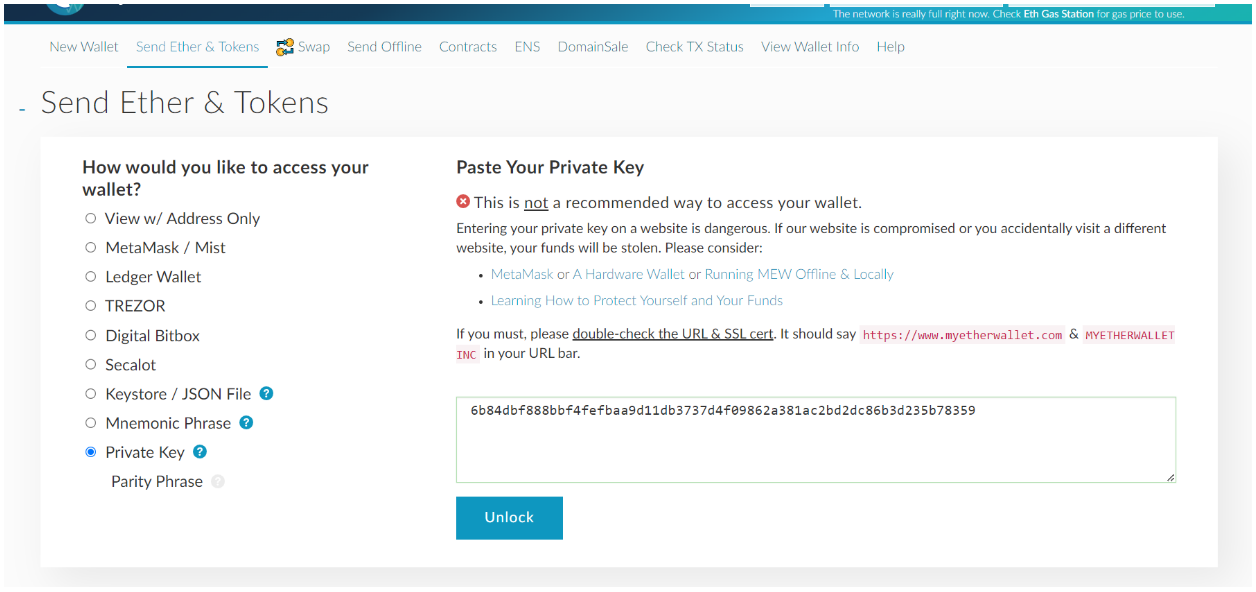


Send Ether & Tokens

How would you like to access your wallet?

- View w/ Address Only
- MetaMask / Mist
- Ledger Wallet
- TREZOR
- Digital Bitbox
- Secalot
- Keystore / JSON File
- Mnemonic Phrase
- Private Key
- Parity Phrase

After pasting the private key:



Send Ether & Tokens

How would you like to access your wallet?

- View w/ Address Only
- MetaMask / Mist
- Ledger Wallet
- TREZOR
- Digital Bitbox
- Secalot
- Keystore / JSON File
- Mnemonic Phrase
- [✓ Private Key]
- Parity Phrase

Paste Your Private Key

This is not a recommended way to access your wallet.

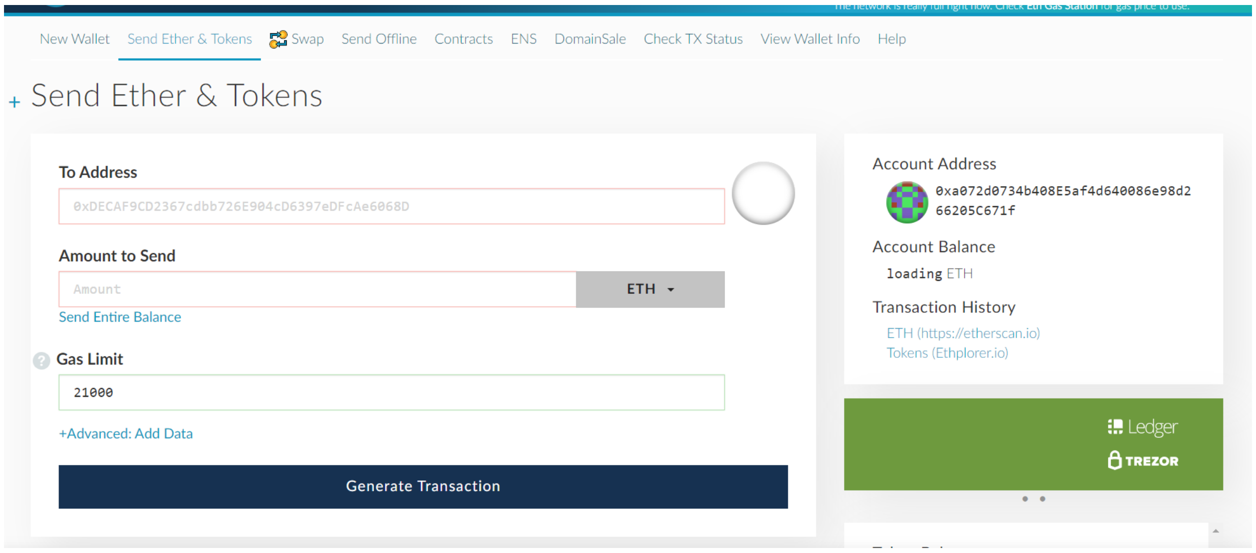
Entering your private key on a website is dangerous. If our website is compromised or you accidentally visit a different website, your funds will be stolen. Please consider:

- MetaMask or A Hardware Wallet or Running MEW Offline & Locally
- Learning How to Protect Yourself and Your Funds

If you must, please double-check the URL & SSL cert. It should say `https://www.myetherwallet.com` & `MYETHERWALLET INC` in your URL bar.

```
6b84dbf888bbf4fefbaa9d11db3737d4f09862a381ac2bd2dc86b3d235b78359
```

After clicking on ‘Unlock’, we get the following screen:



Send Ether & Tokens

To Address: 0xDECAF9CD2367cdbb726E904cD6397eDFcAe6068D

Amount to Send: Amount ETH

Send Entire Balance

Gas Limit: 21000

+Advanced: Add Data

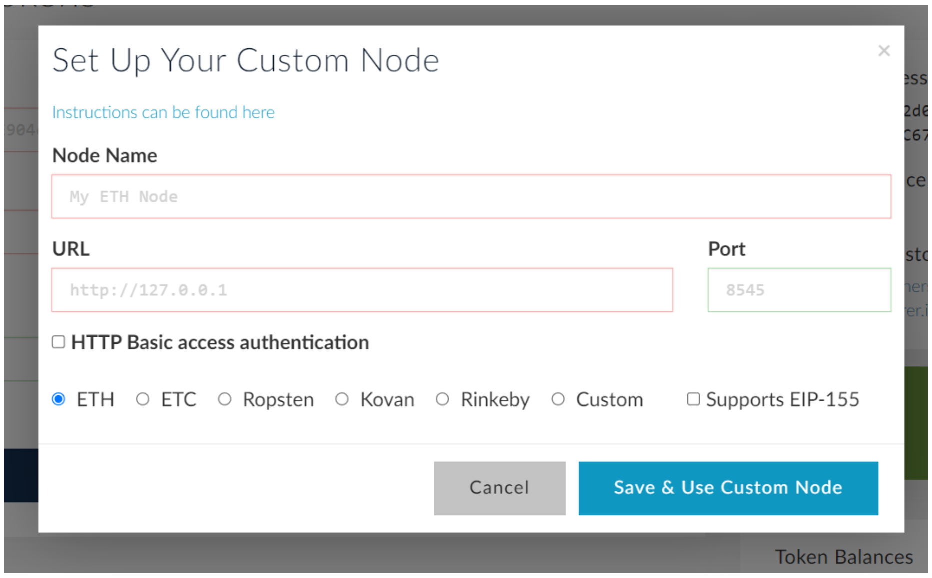
Account Address: 0xa072d0734b408E5af4d640086e98d266205C671f

Account Balance: loading ETH

Transaction History: ETH (https://etherscan.io) Tokens (Ethplorer.io)

2.3:

After clicking on the option to add a custom node:



Set Up Your Custom Node

Instructions can be found here

Node Name: My ETH Node

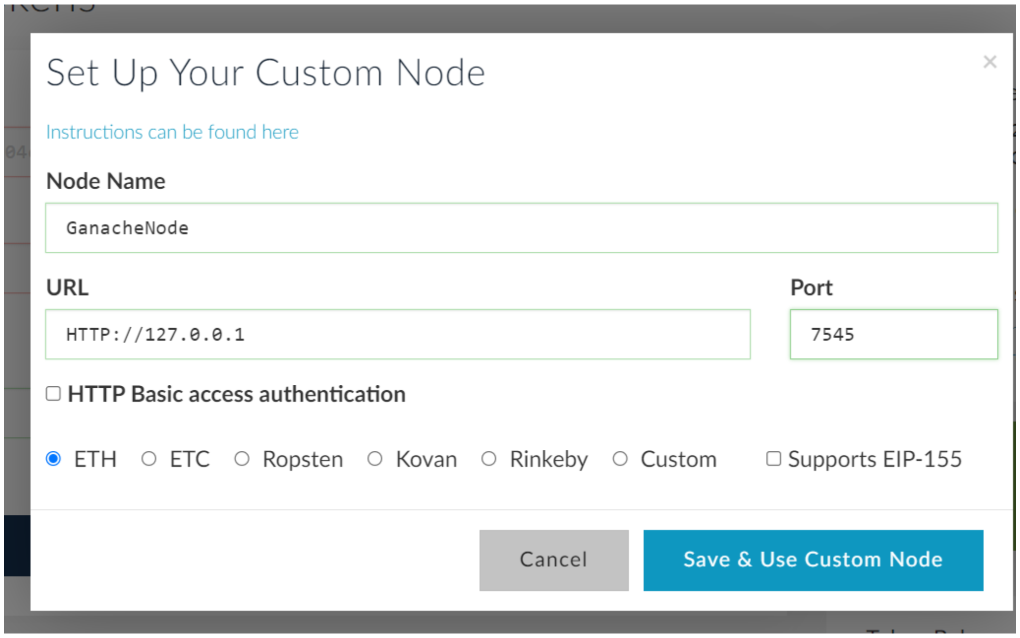
URL: http://127.0.0.1

Port: 8545

☐ HTTP Basic access authentication

[✓ ETH] ETC Ropsten Kovan Rinkeby Custom ☐ Supports EIP-155

After filling in our node localhost address and port:


Set Up Your Custom Node

Instructions can be found here

Node Name

GanacheNode

URL

HTTP://127.0.0.1

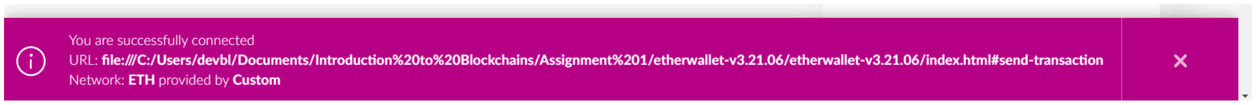
Port

7545

☐ HTTP Basic access authentication

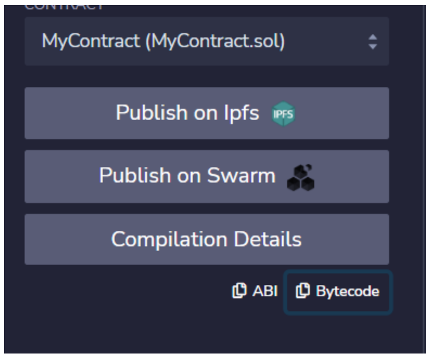
[✓ ETH] ETC Ropsten Kovan Rinkeby Custom ☐ Supports EIP-155

We get the following success message as well:

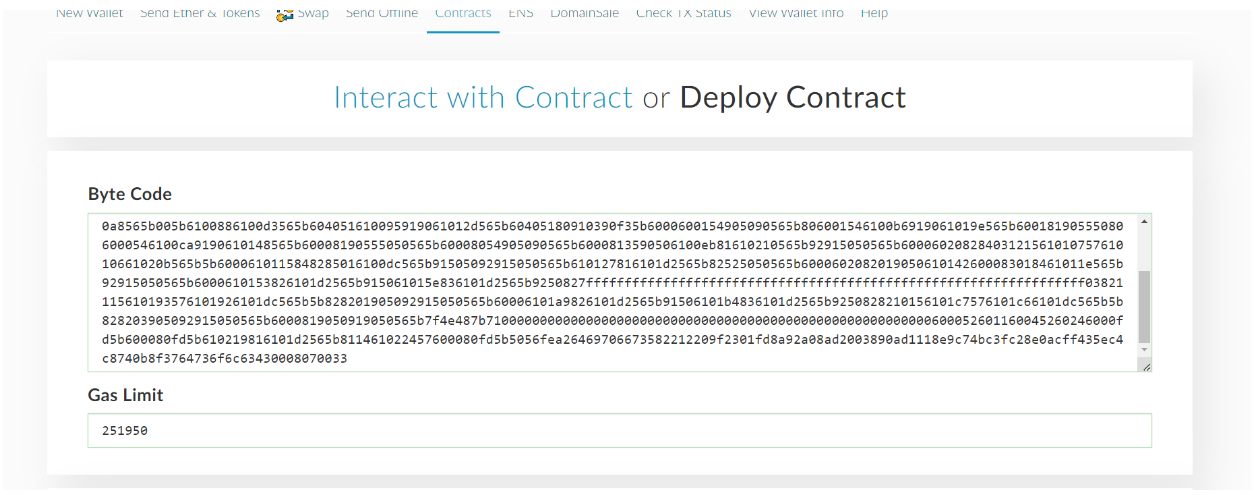

You are successfully connected
URL: **file:///C:/Users/devbl/Documents/Introduction%20to%20Blockchains/Assignment%201/etherwallet-v3.21.06/etherwallet-v3.21.06/index.html#send-transaction**
Network: **ETH** provided by **Custom**

2.4:

We copy the bytecode from the compiled Ethereum contract:


MyContract (MyContract.sol)

After pasting the object section of the bytecode in the MyEtherWallet ‘Interact with Contract or Deploy Contract’ screen:



Interact with Contract or Deploy Contract

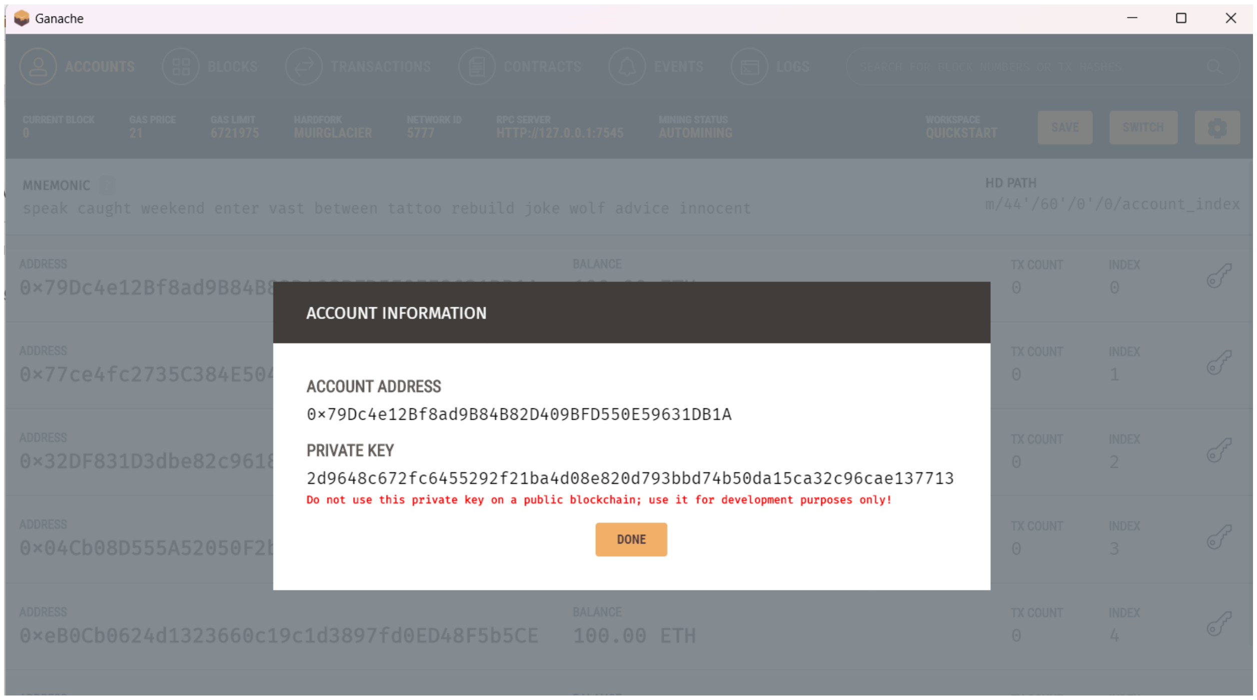
Byte Code

```
0a8565b005b6100886100d3565b604051610095919061012d565b60405180910390f35b6000600154905090565b806001546100b6919061019e565b60018190555080
6000546100ca9190610148565b6000819055505050565b6000805490509056 5b6000813590506100eb81610210565b92915050565b600060208284031215610107576 10
10661020b565b5b6000610115848285016100dc565b91505092915050565b610127816101d2565b82525050565b600060208201905061014260008301846101 1e565b
92915050565b6000610153826101d2565b915061015e836101d2565b9250827fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff03821
115610193576101926101dc565b5b828201905092915050565b60006101a9826101d2565b91506101b4836101d2565b925082821015610 1c7576101c66101dc565b5b
828203905092915050565b6000819050919050565b7f4e487b71000000000000000000000000000000000000000000000000000000006000526011600452602 46000f
d5b600080fd5b610219816101d2565b811461022457600080fd5b5056fea2646970667358221220 9f2301fd8a92a08ad2003890ad1118e9c74bc3fc28e0acff435ec4
c8740b8f3764736f6c63430008070033
```

Gas Limit

```
251950
```

Going back to Ganache, we obtain the details of the #1 wallet:



CURRENT BLOCK 0 GAS PRICE 21 GAS LIMIT 6721975 HARDFORK MUIRGLACIER NETWORK ID 5777 RPC SERVER HTTP://127.0.0.1:7545 MINING STATUS AUTOMINING WORKSPACE QUICKSTART

MNEMONIC
speak caught weekend enter vast between tattoo rebuild joke wolf advice innocent

HD PATH
m/44'/60'/0'/0/account_index

ADDRESS 0×79Dc4e12Bf8ad9B84B8 BALANCE TX COUNT 0 INDEX 0
ADDRESS 0×77ce4fc2735C384E50 TX COUNT 0 INDEX 1
ADDRESS 0×32DF831D3dbe82c961 TX COUNT 0 INDEX 2
ADDRESS 0×04Cb08D555A52050F2 TX COUNT 0 INDEX 3
ADDRESS 0×eB0Cb0624d1323660c19c1d3897fd0ED48F5b5CE BALANCE 100.00 ETH TX COUNT 0 INDEX 4

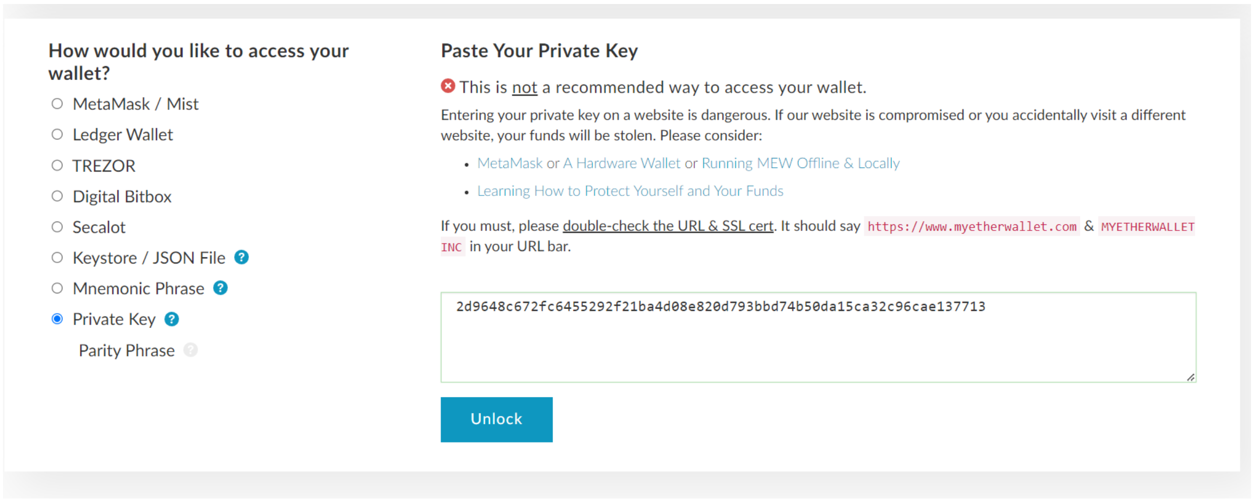
ACCOUNT INFORMATION

ACCOUNT ADDRESS
0×79Dc4e12Bf8ad9B84B82D409BFD550E59631DB1A

PRIVATE KEY
2d9648c672fc6455292f21ba4d08e820d793bbd74b50da15ca32c96cae137713
Do not use this private key on a public blockchain; use it for development purposes only!

DONE

We paste this in the private key section for deploying the smart contract:



How would you like to access your wallet?

- MetaMask / Mist
- Ledger Wallet
- TREZOR
- Digital Bitbox
- Secalot
- Keystore / JSON File
- Mnemonic Phrase
- [✓ Private Key]
- Parity Phrase

Paste Your Private Key

This is not a recommended way to access your wallet.

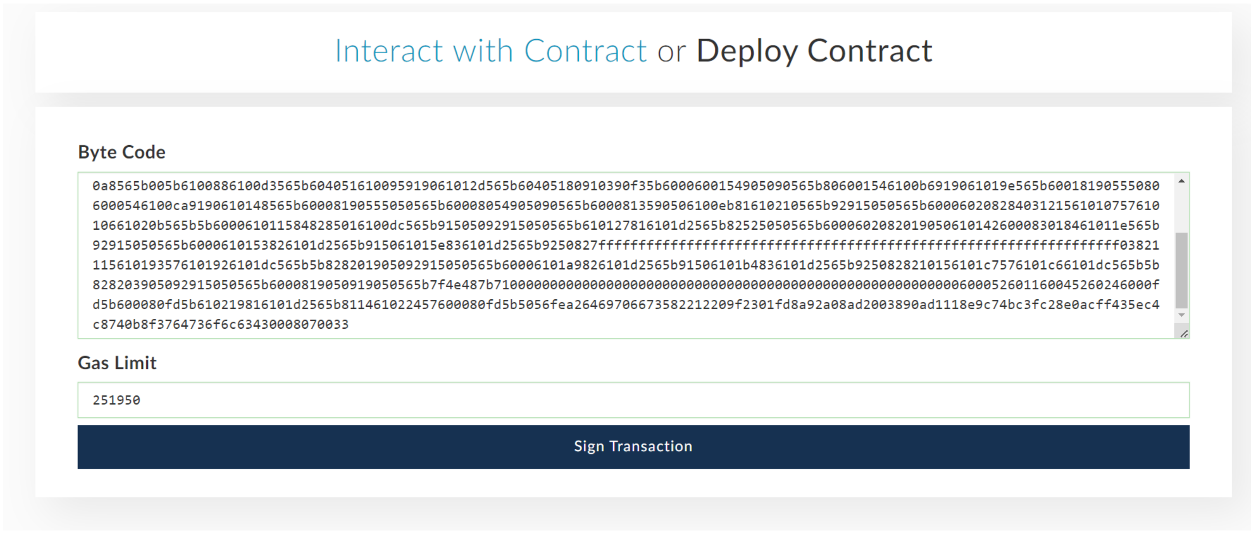
Entering your private key on a website is dangerous. If our website is compromised or you accidentally visit a different website, your funds will be stolen. Please consider:

- MetaMask or A Hardware Wallet or Running MEW Offline & Locally
- Learning How to Protect Yourself and Your Funds

If you must, please double-check the URL & SSL cert. It should say `https://www.myetherwallet.com` & `MYETHERWALLET INC` in your URL bar.

```
2d9648c672fc6455292f21ba4d08e820d793bbd74b50da15ca32c96cae137713
```

After clicking on ‘Unlock’, we get the ‘Sign Transaction’ screen:



Interact with Contract or Deploy Contract

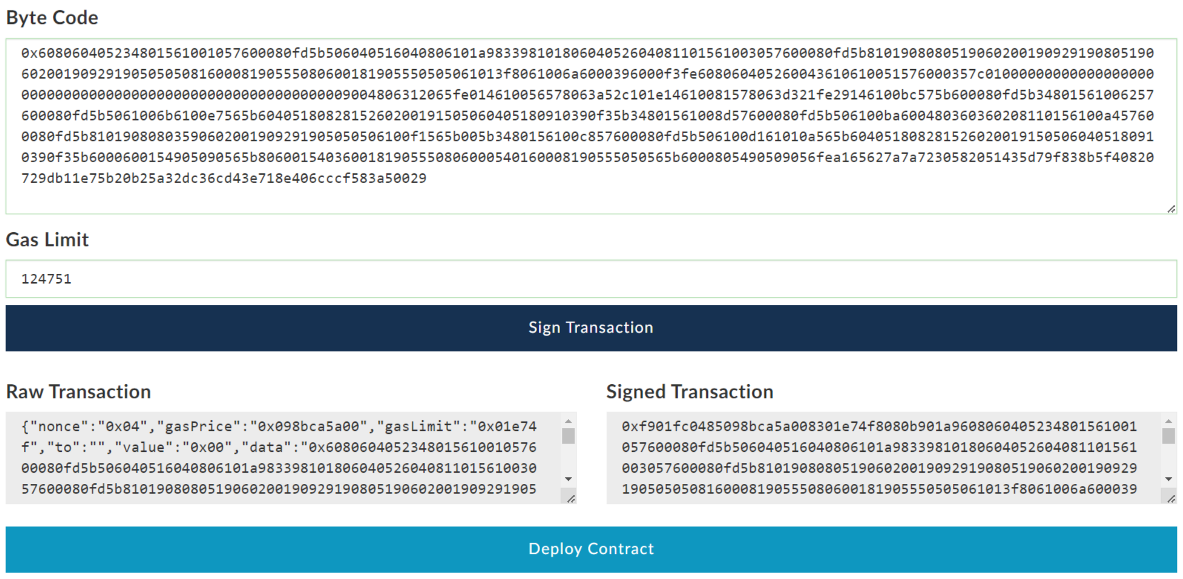
Byte Code

```
0a8565b005b6100886100d3565b604051610095919061012d565b60405180910390f35b6000600154905090565b806001546100b6919061019e565b60018190555080
6000546100ca9190610148565b60008190555050565b6000805490509056 5b600081359050 6100eb81610210565b92915050565b6000602082840312156101075761 0
10661020b565b5b600061011584828501 6100dc565b91505092915050565b610127816101d2565b82525050565b600060208201905061014260008301846101 1e565b
92915050565b6000610153826101d2565b915061015e836101d2565b9250827fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff03821
1156101935761019261 01dc565b5b828201905092915050565b60006101a9826101d2565b91506101b4836101d2565b925082821015 6101c7576101c66101dc565b5b
828203905092915050565b6000819050919050565b7f4e487b71000000000000000000000000000000000000000000000000000000006000526011600452602460 00f
d5b600080fd5b610219816101d2565b811461022457600080fd5b5056fea2646970667358221220 9f2301fd8a92a08ad2003890ad1118e9c74bc3fc28e0acff435ec4
c8740b8f3764736f6c63430008070033
```

Gas Limit

```
251950
```

After signing, we get the raw transaction and signed transaction data:



Byte Code

```
0x608060405234801561001057600080fd5b506040516040806101a98339810180604052604081101561003057600080fd5b8101908080519060200190929190805190
602001909291905050508160008190555080600181905550505061013f8061006a6000396000f3fe608060405260043610610051576000357c0100000000000000000000
00000000000000000000000000000000000000009004806312065fe014610056578063a52c101e14610081578063d321fe29146100bc575b600080fd5b34801561006257
600080fd5b5061006b6100e7565b6040518082815260200191505060405180910390f35b34801561008d57600080fd5b506100ba6004803603602081101561 00a45760
0080fd5b8101908080359060200190929190505050610 0f1565b005b3480156100c857600080fd5b506100d161010a565b60405180828152602001915050604051 8091
0390f35b6000600154905090565b8060015403600181905550806000540160008190555050565b6000805490509056fea165627a7a7230582051435d79f838b5f40820
729db11e75b20b25a32dc36cd43e718e406cccf583a50029
```

Gas Limit

```
124751
```

Sign Transaction

Raw Transaction

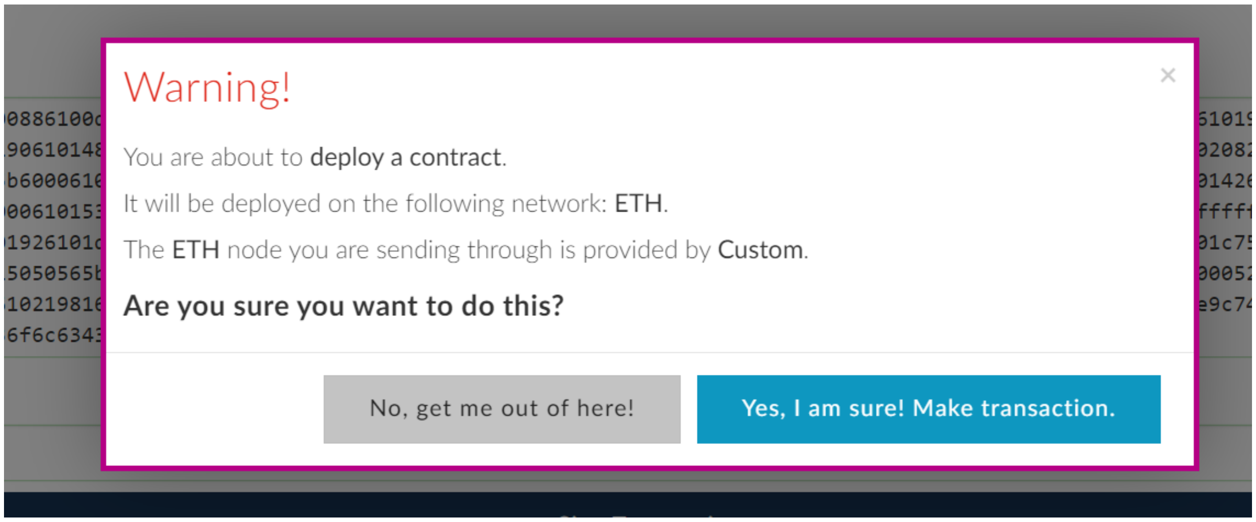
```
{"nonce":"0x04","gasPrice":"0x098bca5a00","gasLimit":"0x01e74
f","to":"","value":"0x00","data":"0x608060405234801561001057б
00080fd5b506040516040806101a98339810180604052604081101561003O
57600080fd5b8101908080519060200190929190805190602001909291905
```

Signed Transaction

```
0xf901fc0485098bca5a008301e74f8080b901a9608060405234801561001
057600080fd5b506040516040806101a98339810180604052604081101561
003057600080fd5b810190808051906020019092919080519060200190929
1905050508160008190555080600181905550505061013f8061006a600039
```

Deploy Contract

On clicking ‘Deploy Contract’, we get this screen:



Warning!

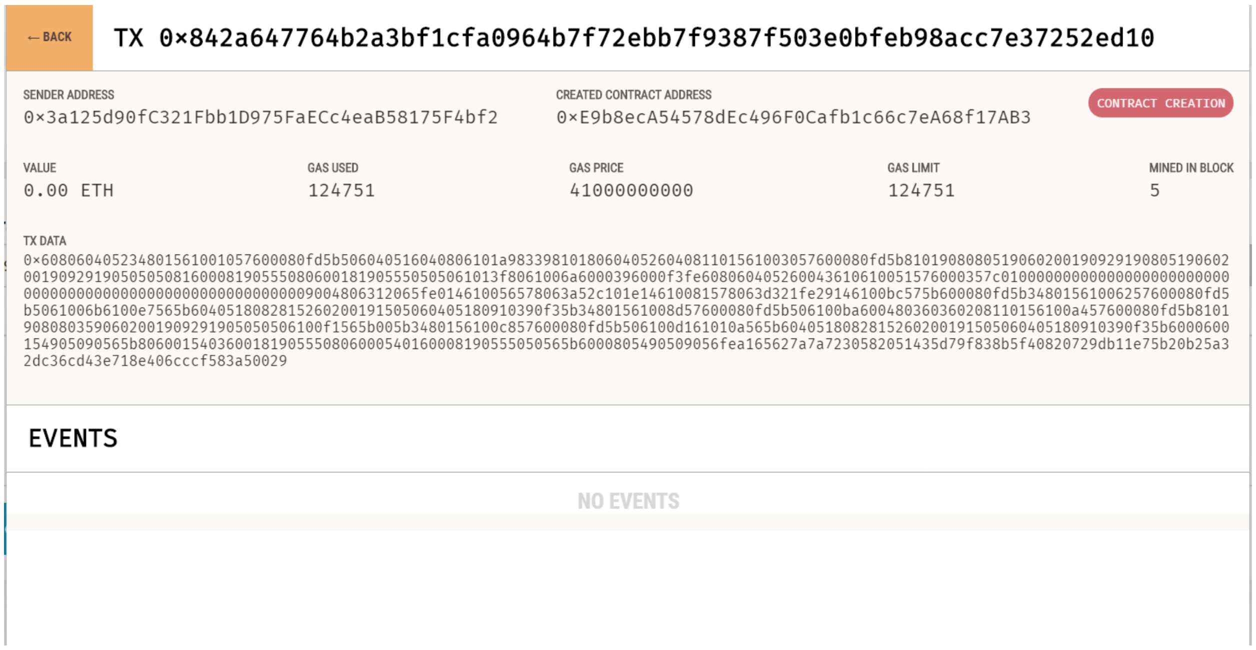
You are about to **deploy a contract**.

It will be deployed on the following network: **ETH**.

The **ETH** node you are sending through is provided by **Custom**.

Are you sure you want to do this?

After deploying the contract and going to Transactions tab in Ganache, we can find the smart contract creation transaction:

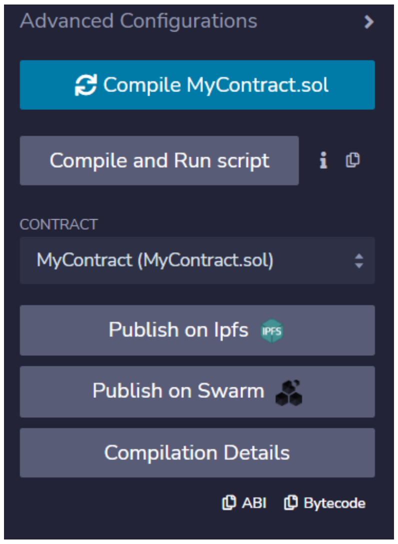


```
TX 0×842a647764b2a3bf1cfa0964b7f72ebb7f9387f503e0bfeb98acc7e37252ed10

SENDER ADDRESS                               CREATED CONTRACT ADDRESS
0×3a125d90fC321Fbb1D975FaECc4eaB58175F4bf2   0×E9b8ecA54578dEc496F0Cafb1c66c7eA68f17AB3

VALUE          GAS USED       GAS PRICE        GAS LIMIT        MINED IN BLOCK
0.00 ETH       124751         41000000000      124751           5

TX DATA
0×608060405234801561001057600080fd5b5060405160408061 01a98339810180604052604081101561003057600080fd5b8101908080519060200190929190805190602
00190929190505050816000819055508060018190555050506101 3f8061006a6000396000f3fe608060405260043610610051576000357c0100000000000000000000000000
0000000000000000000000000000000009004806312065fe014610056578063a52c101e14610081578063d321fe29146100bc575b600080fd5b34801561006257600080fd5
b5061006b6100e7565b6040518082815260200191505060405180910390f35b34801561008d57600080fd5b506100ba600480360360208110156100a457600080fd5b8101
9080803590602001909291905050506100f1565b005b3480156100c857600080fd5b506100d161010a565b6040518082815260200191505060405180910390f35b6000600
154905090565b8060015403600181905550806000540160008190555050565b6000805490509056fea165627a7a7230582051435d79f838b5f40820729db11e75b20b25a3
2dc36cd43e718e406cccf583a50029
```

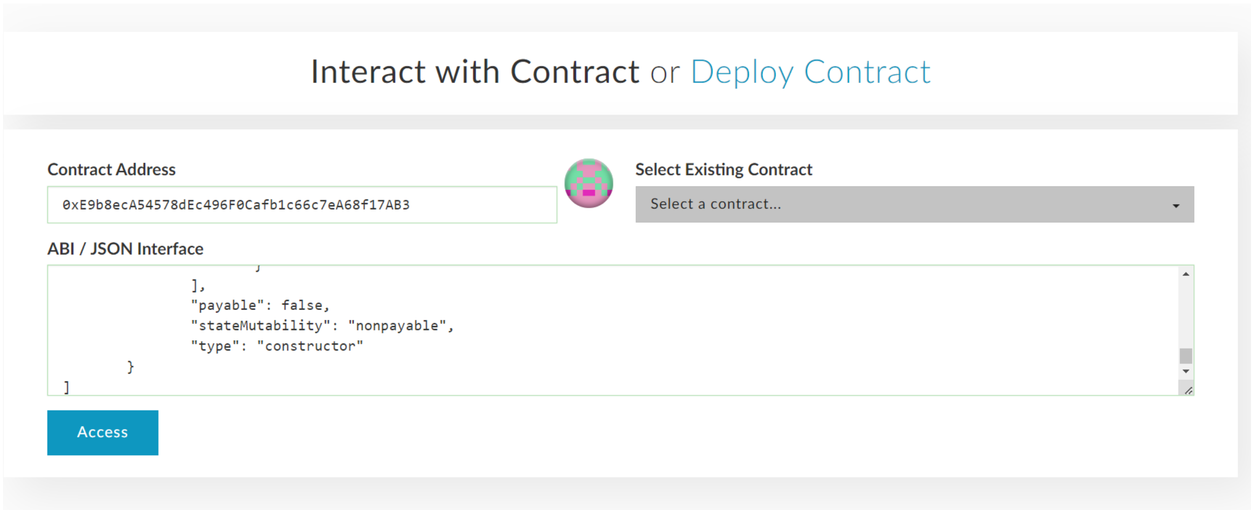
The mined block is displayed as 5 because my past tries at deploying the contract were not successful, due to some gas issues.

2.5:

In remix, we can find the ABI copy button here:



Now, after copying the ABI by clicking on the button, we go the MyEtherWallet screen and paste the details, which are the Contract Address and Contract ABI:


Interact with Contract or Deploy Contract

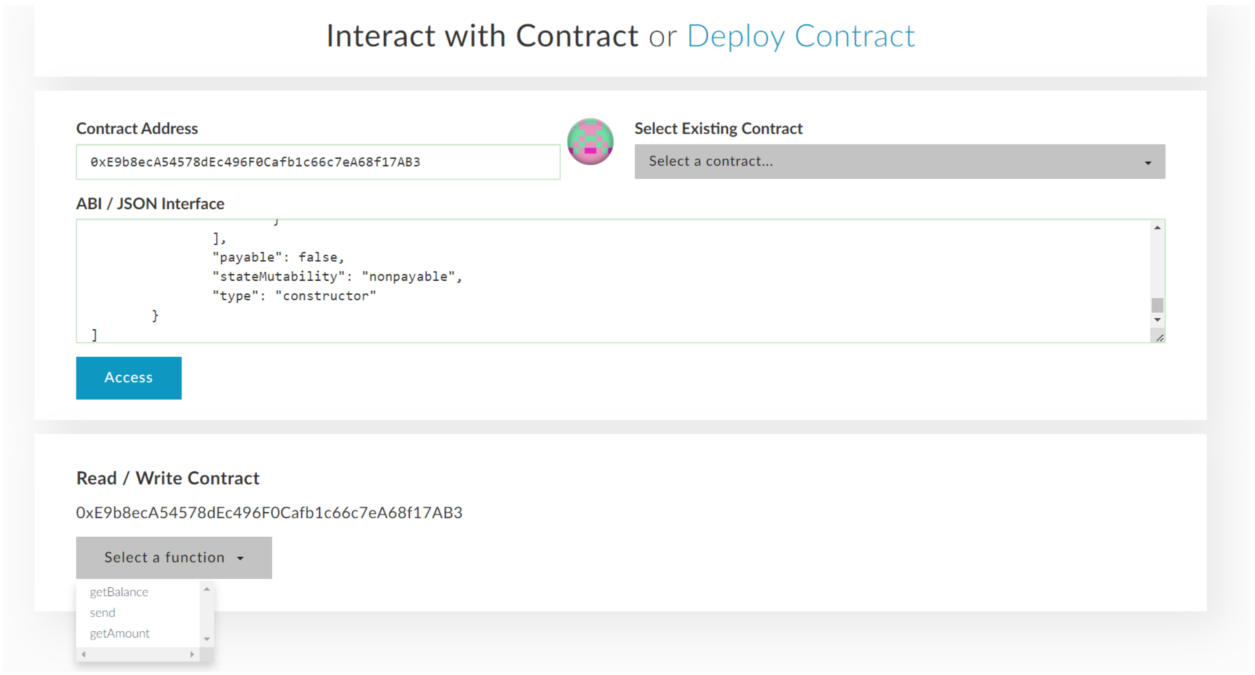
Contract Address
0xE9b8ecA54578dEc496F0Cafb1c66c7eA68f17AB3

Select Existing Contract
Select a contract...

ABI / JSON Interface

```
            ],
            "payable": false,
            "stateMutability": "nonpayable",
            "type": "constructor"
        }
]
```

On clicking the ‘Access’ button, we can see functions within the smart contract which we can access:


Interact with Contract or Deploy Contract

Contract Address
0xE9b8ecA54578dEc496F0Cafb1c66c7eA68f17AB3

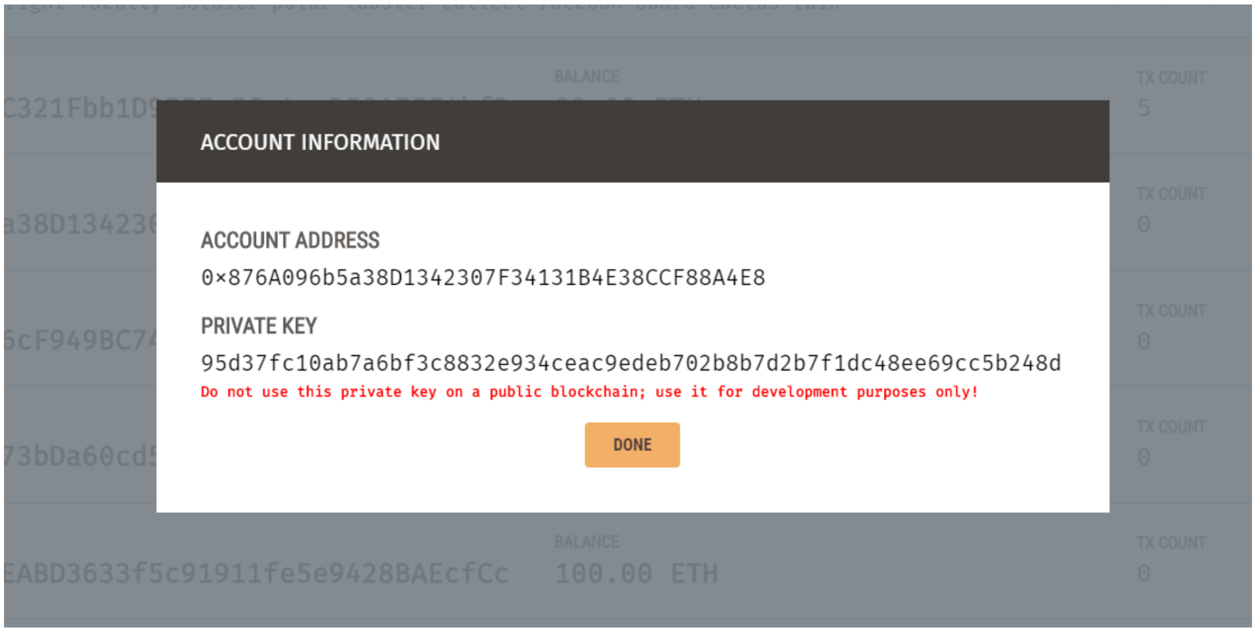
Select Existing Contract
Select a contract...

ABI / JSON Interface

```
            ],
            "payable": false,
            "stateMutability": "nonpayable",
            "type": "constructor"
        }
]
```

Read / Write Contract
0xE9b8ecA54578dEc496F0Cafb1c66c7eA68f17AB3

Now, we go to Ganache and select details for User #2:

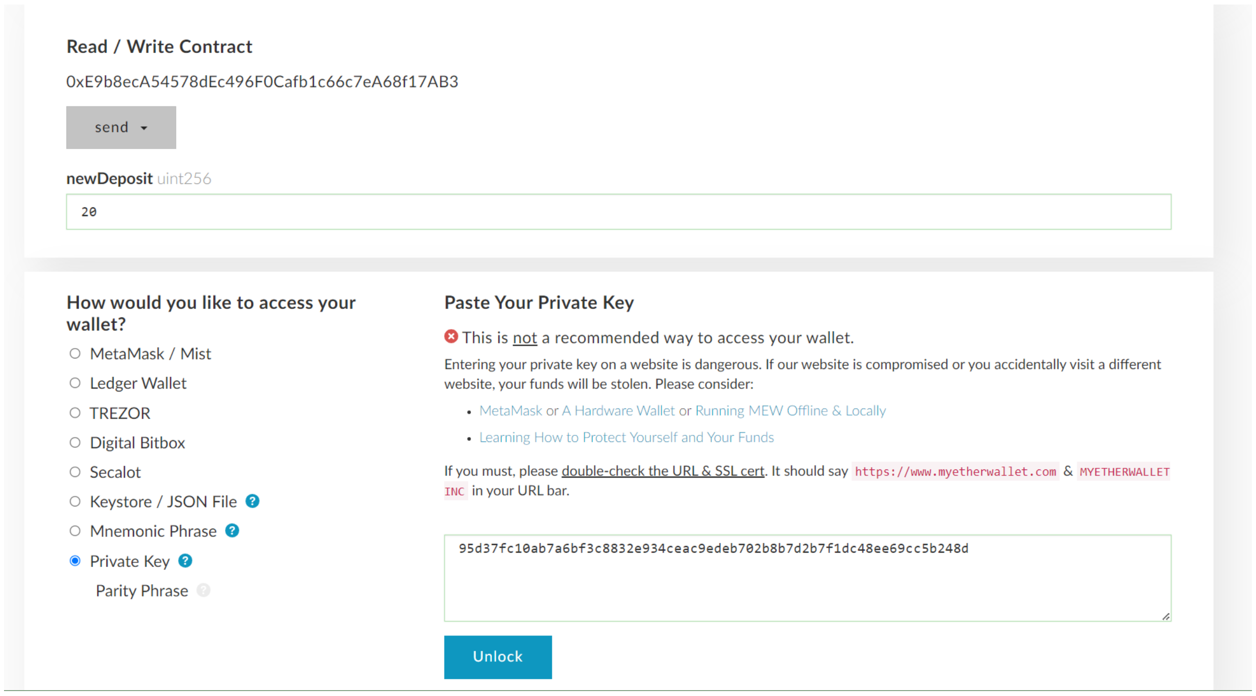

ACCOUNT INFORMATION

ACCOUNT ADDRESS
0×876A096b5a38D1342307F34131B4E38CCF88A4E8

PRIVATE KEY
95d37fc10ab7a6bf3c8832e934ceac9edeb702b8b7d2b7f1dc48ee69cc5b248d

Do not use this private key on a public blockchain; use it for development purposes only!

After filling 20 ETH in the amount to be sent field and entering the private key of account #2 from Ganache:


Read / Write Contract

0xE9b8ecA54578dEc496F0Cafb1c66c7eA68f17AB3

send

newDeposit uint256

20

How would you like to access your wallet?

- MetaMask / Mist
- Ledger Wallet
- TREZOR
- Digital Bitbox
- Secalot
- Keystore / JSON File
- Mnemonic Phrase
- [✓ Private Key]
- Parity Phrase

Paste Your Private Key

This is not a recommended way to access your wallet.

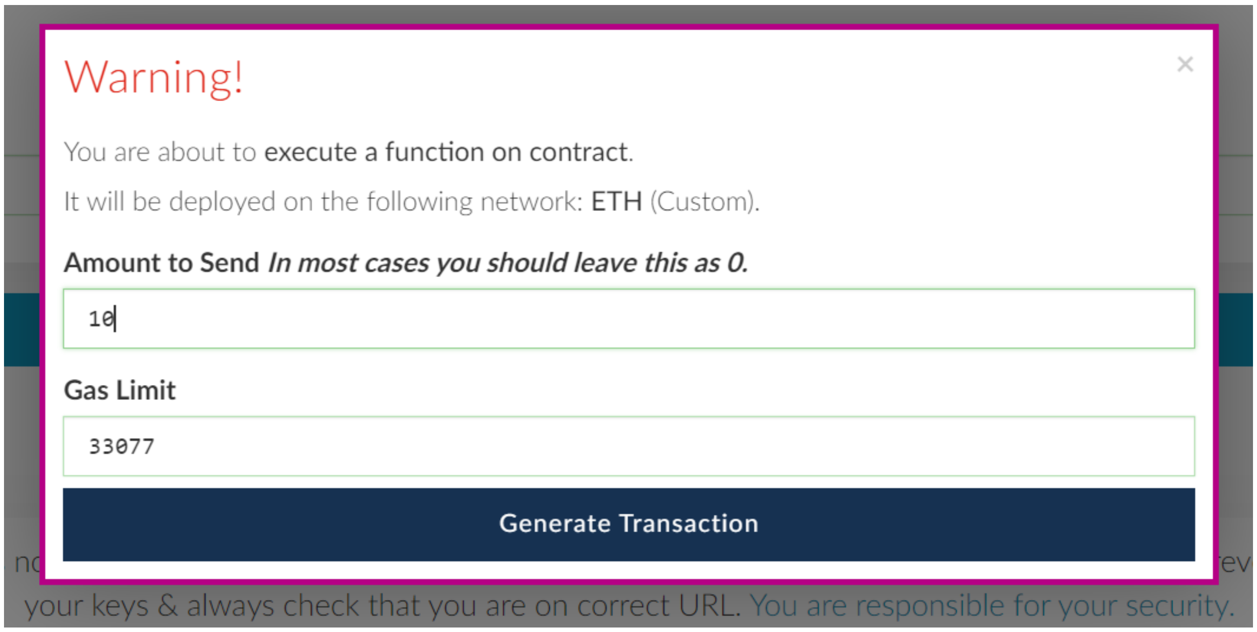
Entering your private key on a website is dangerous. If our website is compromised or you accidentally visit a different website, your funds will be stolen. Please consider:

- MetaMask or A Hardware Wallet or Running MEW Offline & Locally
- Learning How to Protect Yourself and Your Funds

If you must, please double-check the URL & SSL cert. It should say `https://www.myetherwallet.com` & `MYETHERWALLET INC` in your URL bar.

95d37fc10ab7a6bf3c8832e934ceac9edeb702b8b7d2b7f1dc48ee69cc5b248d

After unlocking and signing transaction to get raw transaction data and signed transaction data:



Warning!

You are about to execute a function on contract.

It will be deployed on the following network: ETH (Custom).

Amount to Send *In most cases you should leave this as 0.*

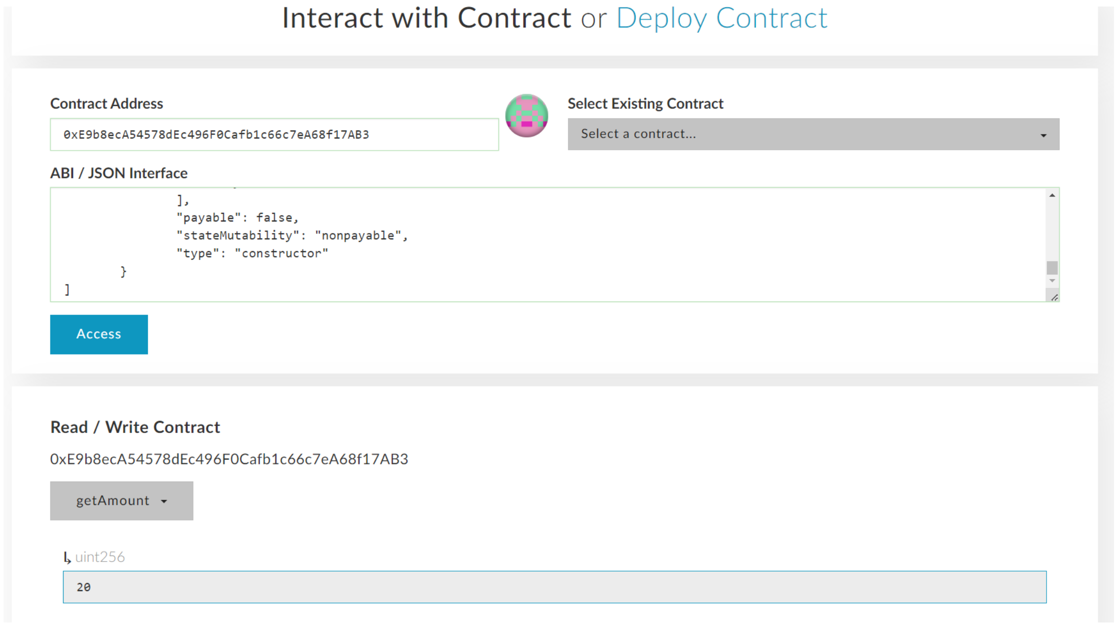
10

Gas Limit

33077

your keys & always check that you are on correct URL. You are responsible for your security.

Now, after making the transaction, we click on getAmount from the function dropdown menu:


Interact with Contract or Deploy Contract

Contract Address

```
0xE9b8ecA54578dEc496F0Cafb1c66c7eA68f17AB3
```

Select Existing Contract

Select a contract...

ABI / JSON Interface

```
                ],
                "payable": false,
                "stateMutability": "nonpayable",
                "type": "constructor"
        }
]
```

Read / Write Contract

0xE9b8ecA54578dEc496F0Cafb1c66c7eA68f17AB3

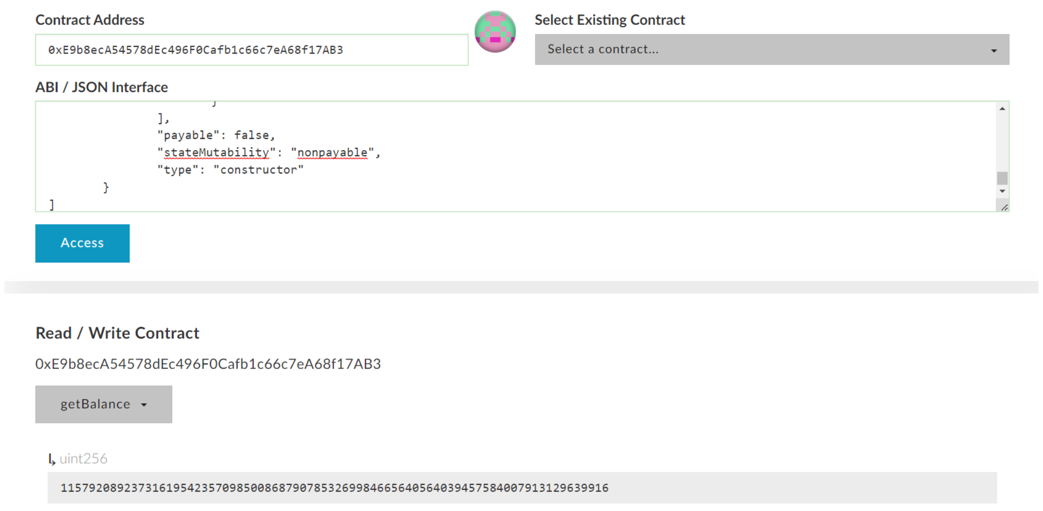
getAmount

↳ uint256

```
20
```

The field immediately shows as 20, since it is a view function and does not require a transaction to be made from our account. It shows 20 since we entered ‘20’ as ‘newDeposit’.

For some reason, the ‘getBalance’ function returns this long value:


Contract Address

```
0xE9b8ecA54578dEc496F0Cafb1c66c7eA68f17AB3
```

Select Existing Contract

Select a contract...

ABI / JSON Interface

```
                ],
                "payable": false,
                "stateMutability": "nonpayable",
                "type": "constructor"
        }
]
```

Read / Write Contract

0xE9b8ecA54578dEc496F0Cafb1c66c7eA68f17AB3

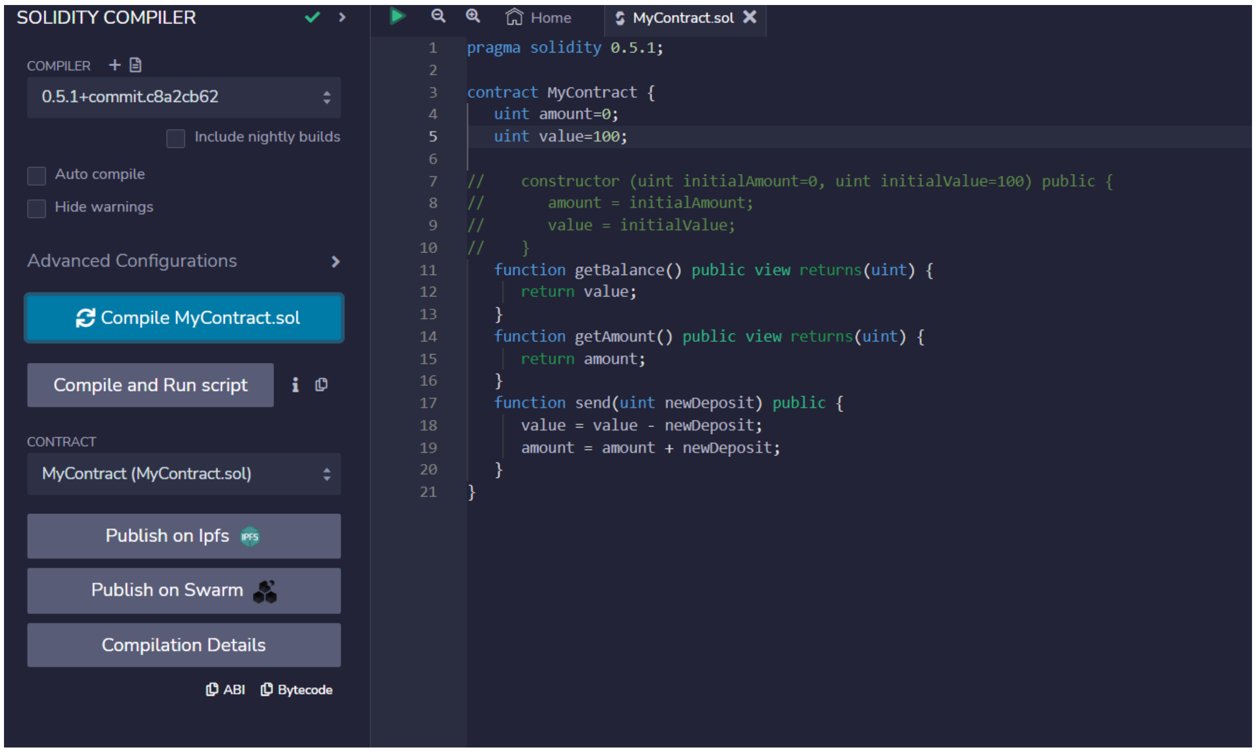
getBalance

↳ uint256

```
115792089237316195423570985008687907853269984665640564039457584007913129639916
```

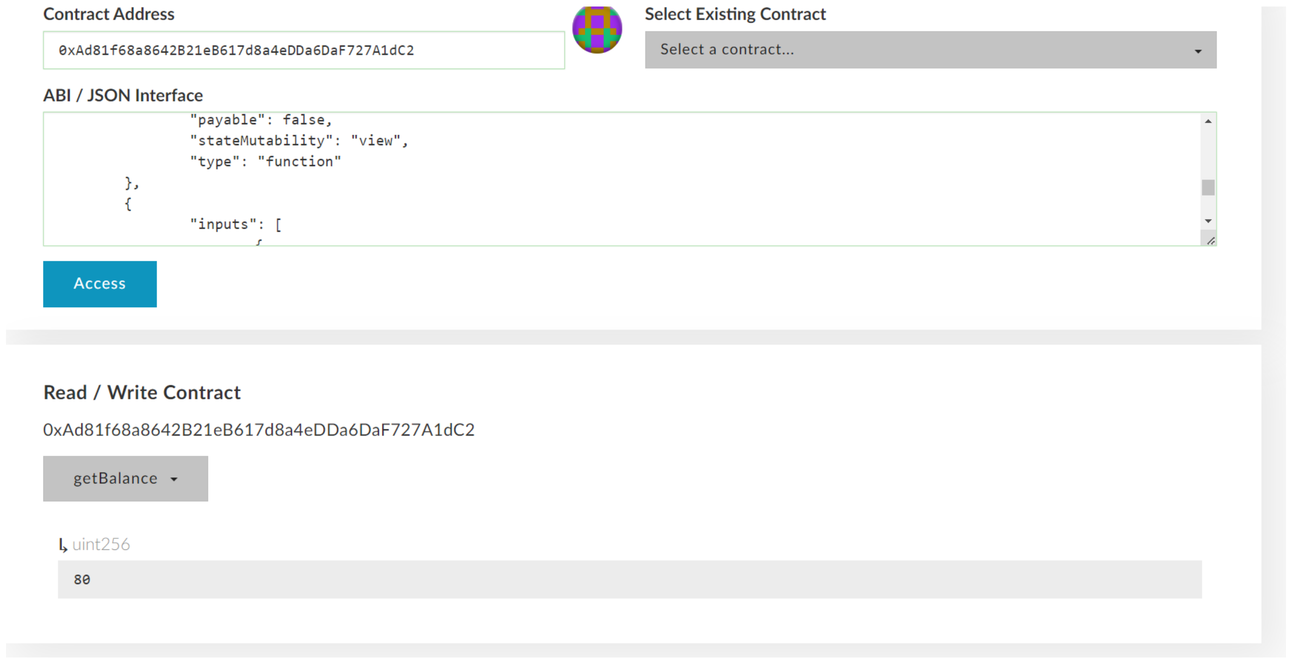
This is because we did not pass any values to the constructor while deploying to the blockchain. Since solidity by default initializes uint to 0, while we invoke the send() function with 10, it malfunctions and makes the ‘value’ variable ‘the maximum value of uint256 - whatever value we sent’.

I was able to find out that MyEtherWallet does not allow passing arguments to the constructor and we must hardcode the bytecode for doing so. An easy way to bypass this would be to hardcode the variables like this:



```
pragma solidity 0.5.1;

contract MyContract {
    uint amount=0;
    uint value=100;

//    constructor (uint initialAmount=0, uint initialValue=100) public {
//        amount = initialAmount;
//        value = initialValue;
//    }
    function getBalance() public view returns(uint) {
        return value;
    }
    function getAmount() public view returns(uint) {
        return amount;
    }
    function send(uint newDeposit) public {
        value = value - newDeposit;
        amount = amount + newDeposit;
    }
}
```

Redeploying contract, and going through the whole process of sending ‘20’, we get the following correct value:



```
0xAd81f68a8642B21eB617d8a4eDDa6DaF727A1dC2
```

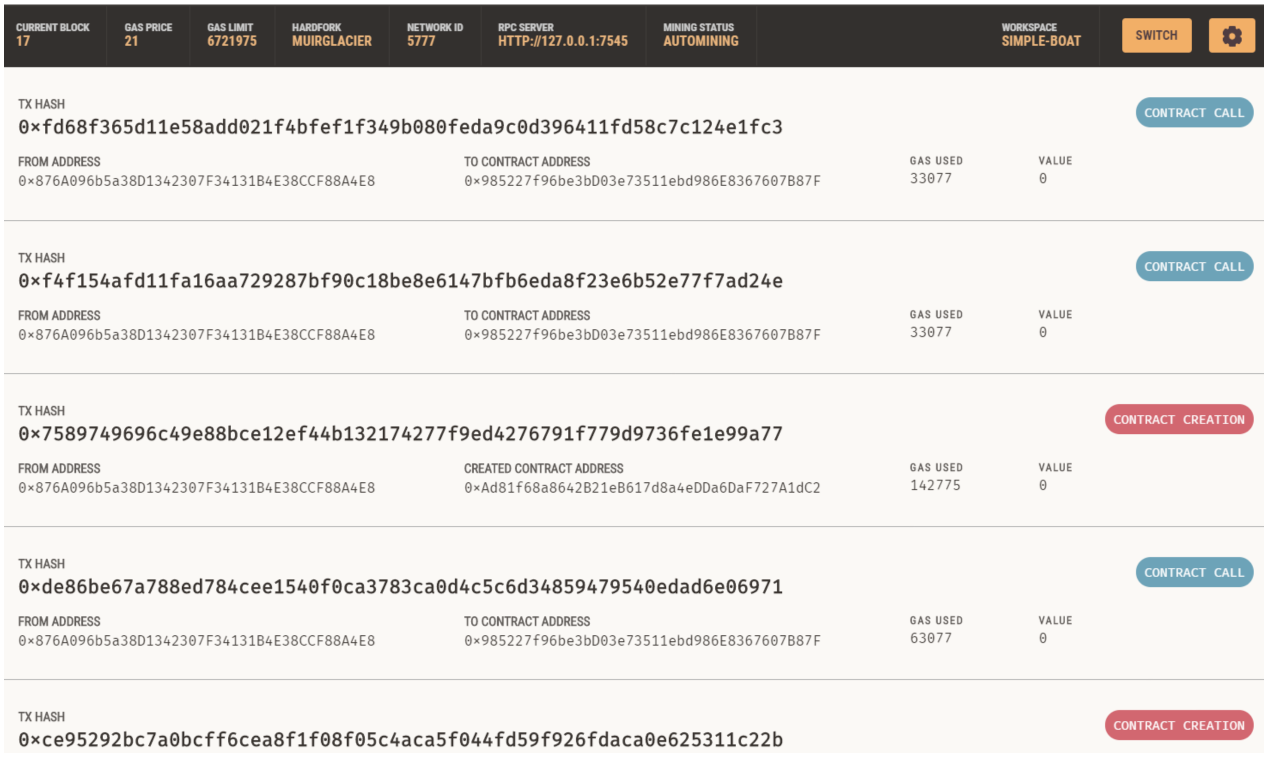
```
                "payable": false,
                "stateMutability": "view",
                "type": "function"
        },
        {
                "inputs": [
```

0xAd81f68a8642B21eB617d8a4eDDa6DaF727A1dC2

```
80
```

Another way to do this without hardcoding the variables and by using the constructor would be to make a Truffle project, connect it with Ganache and pass values to the constructor.

Now, the transactions:



```
CURRENT BLOCK  GAS PRICE  GAS LIMIT  HARDFORK      NETWORK ID  RPC SERVER              MINING STATUS        WORKSPACE
17             21         6721975    MUIRGLACIER   5777        HTTP://127.0.0.1:7545   AUTOMINING           SIMPLE-BOAT    SWITCH

TX HASH                                                                                           CONTRACT CALL
0×fd68f365d11e58add021f4bfef1f349b080feda9c0d396411fd58c7c124e1fc3
FROM ADDRESS                                  TO CONTRACT ADDRESS                           GAS USED   VALUE
0×876A096b5a38D1342307F34131B4E38CCF88A4E8    0×985227f96be3bD03e73511ebd986E8367607B87F    33077      0

TX HASH                                                                                           CONTRACT CALL
0×f4f154afd11fa16aa729287bf90c18be8e6147bfb6eda8f23e6b52e77f7ad24e
FROM ADDRESS                                  TO CONTRACT ADDRESS                           GAS USED   VALUE
0×876A096b5a38D1342307F34131B4E38CCF88A4E8    0×985227f96be3bD03e73511ebd986E8367607B87F    33077      0

TX HASH                                                                                           CONTRACT CREATION
0×7589749696c49e88bce12ef44b132174277f9ed4276791f779d9736fe1e99a77
FROM ADDRESS                                  CREATED CONTRACT ADDRESS                      GAS USED   VALUE
0×876A096b5a38D1342307F34131B4E38CCF88A4E8    0×Ad81f68a8642B21eB617d8a4eDDa6DaF727A1dC2    142775     0

TX HASH                                                                                           CONTRACT CALL
0×de86be67a788ed784cee1540f0ca3783ca0d4c5c6d34859479540edad6e06971
FROM ADDRESS                                  TO CONTRACT ADDRESS                           GAS USED   VALUE
0×876A096b5a38D1342307F34131B4E38CCF88A4E8    0×985227f96be3bD03e73511ebd986E8367607B87F    63077      0

TX HASH                                                                                           CONTRACT CREATION
0×ce95292bc7a0bcff6cea8f1f08f05c4aca5f044fd59f926fdaca0e625311c22b
```

As we can see, there are many transactions- Contract creations and contract calls, since I was trying out various methods and exploring how to use Ganache and MyEtherWallet in general.