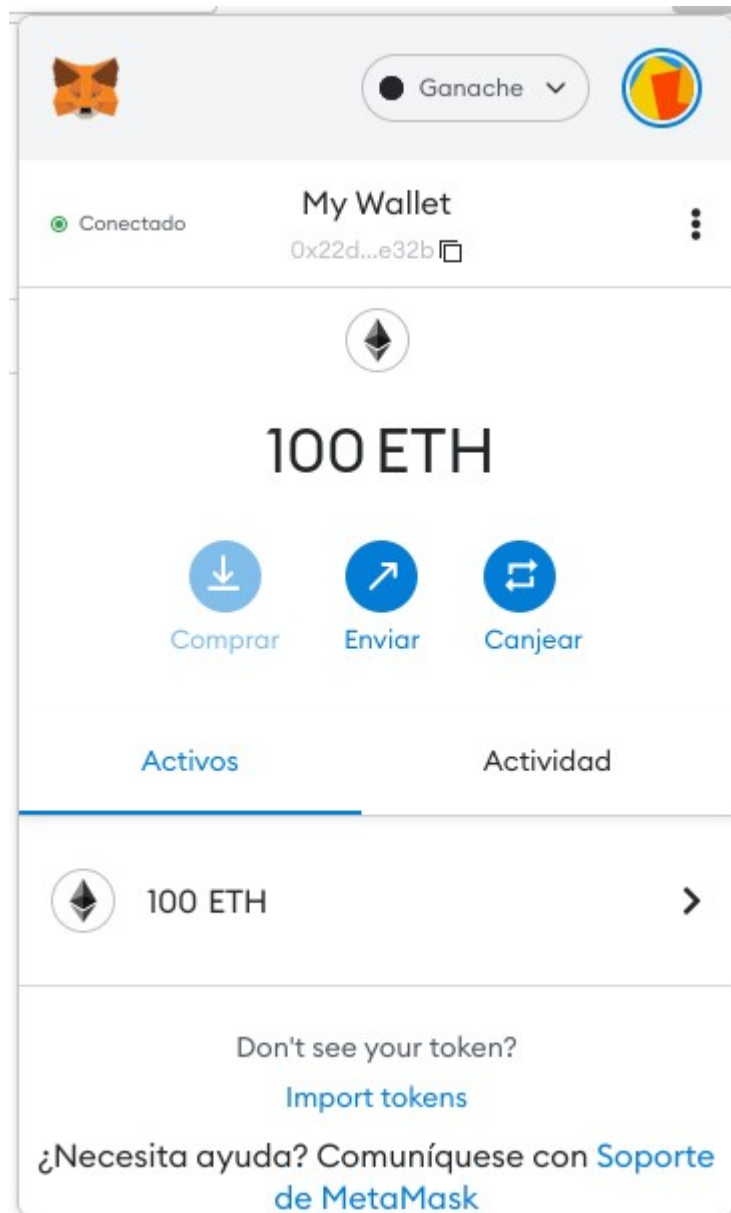
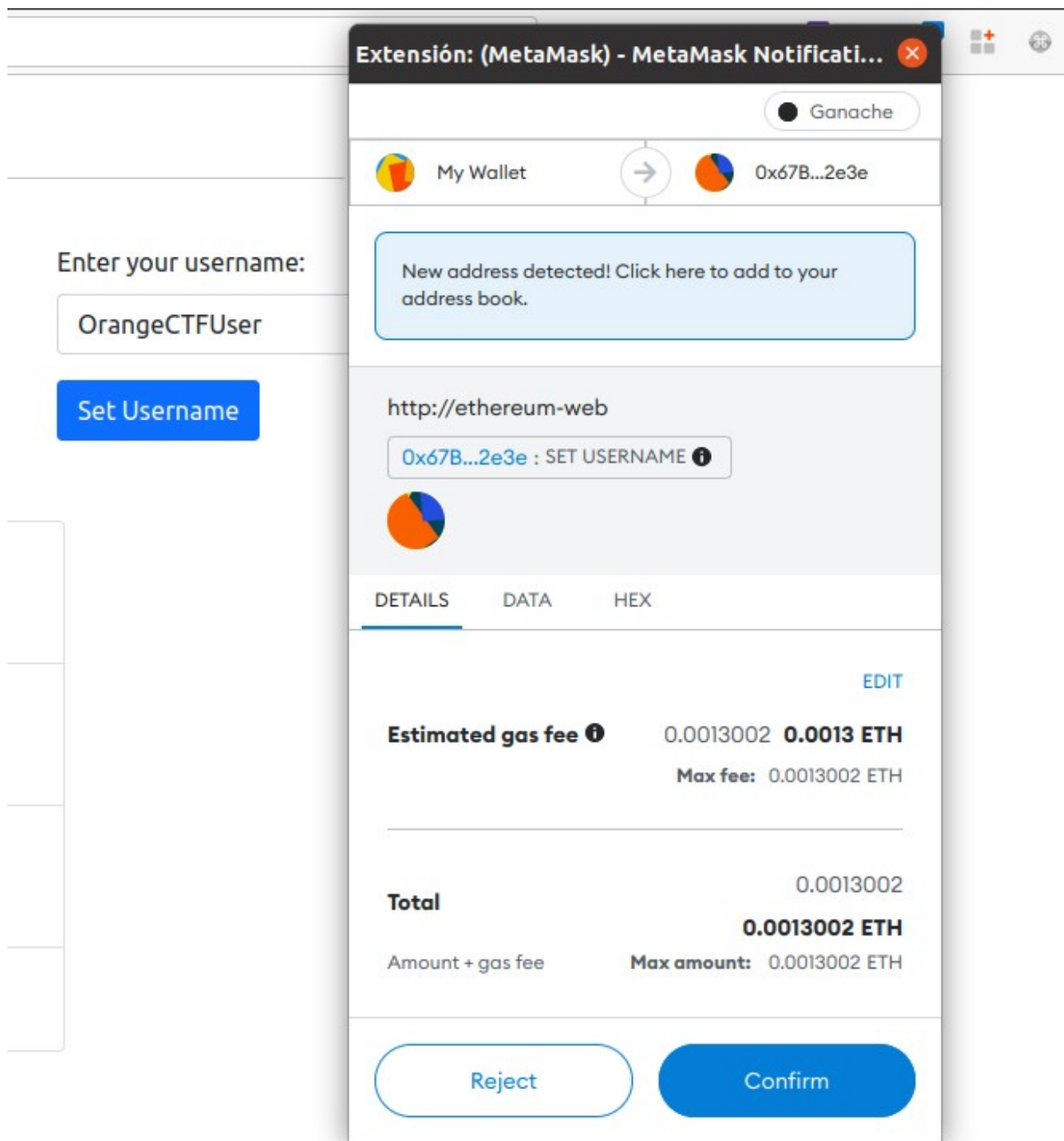


# Smart Contracts Manual

First time you enter the web application you should see that you have a wallet in Metamask with 100 ETH, and you are connected to the application.

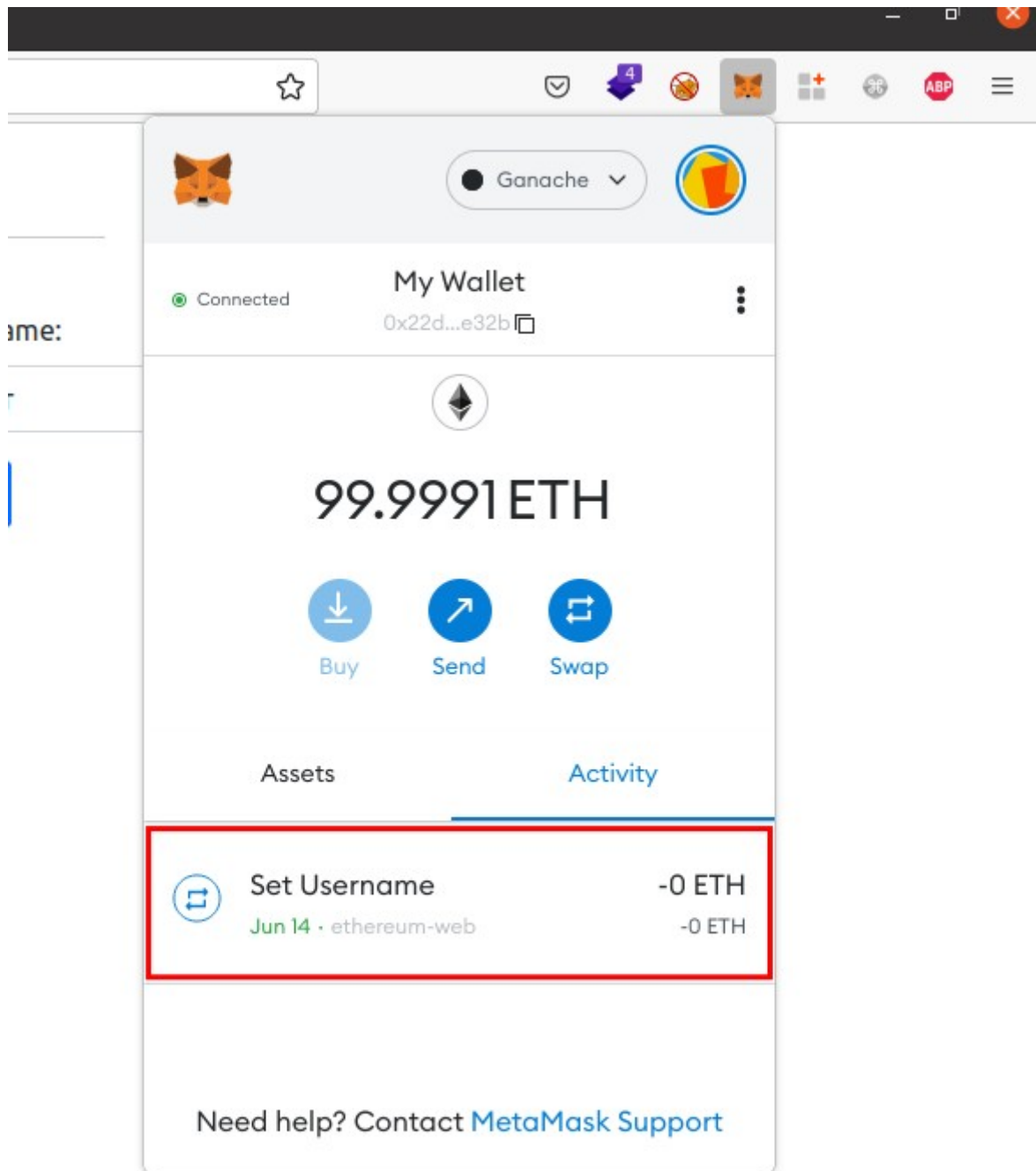


You can set your name, just to test the interaction with the blockchain, although this is not necessary.

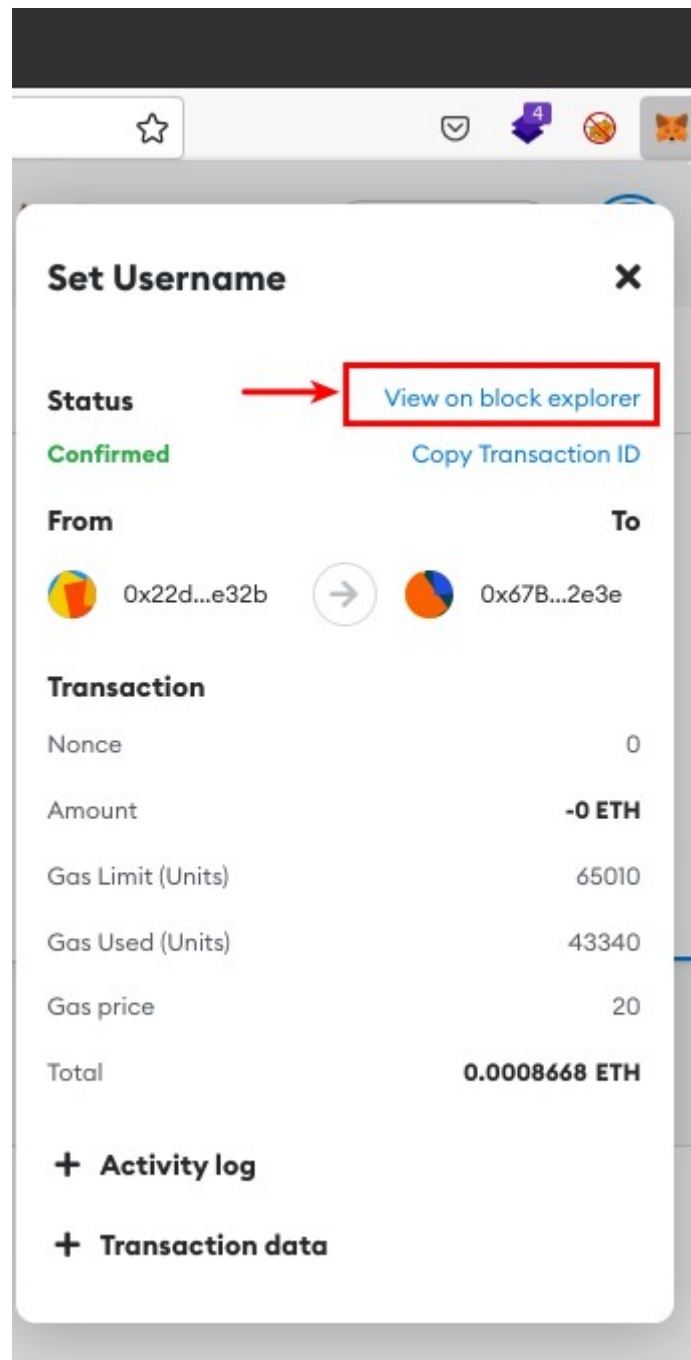


We are executing a function on a Smart Contract in the Blockchain to perform this operation. Every operation consumes a gas fee, so you would have to pay a small amount of ether.

The operation has been registered in our accounting ledger:



If you click on the operation block, you can see more details about it:



Now you can click on “View on block explorer” and you will see the transaction on the public block explorer.

Transaction Details

Transaction Hash

0x50caedc4f0543f5f2ed210f8320b8eac0ee17e1dacac4b7c65d52c7bdcef366f

Result

Success

Status

Confirmed

Confirmed by 1 block

Block

72

Timestamp

16 minutes ago | June-14-2022 11:38:58 AM +2 UTC | Confirmed within <= 0.0 seconds

From

0x22d491bde2303f2f43325b2108d26f1eaba1e32b

Interacted With (To)

0x67b5656d60a809915323bf2c40a8bef15a152e3e

Value

0 Ether (\$0.00 USD)

Transaction Fee

0.0008668 Ether (\$1.05 USD)

Gas Price

20 Gwei

Gas Limit

65,010

Gas Used by Transaction

43,340 | 66.67%

Nonce

Position

0

0

Raw Input

UTF-8

0x1:

OrangeCTFUser

Back on the Challenge Board, there is a requirement to deploy the first challenge: pay 1 ether (this amount will come back to you if you succeed in solving the challenge).

Challenges Board

Welcome back, OrangeCTFUser!

Challenge 1: Lottery	The Bank is offering a price for a lottery game. Would you be able to guess the winner number? You only need to pay 1 eth to participate!	Deploy
Challenge 2: Token Sale	Don't you have Bank tokens yet? This coin will allow you to do a lot of things! Please, enter here to exchange eth for Bank Coins. (You need to pay 1 eth to use this service)	Deploy
Challenge 3: Retirement Fund	If you want to invest your money in a safe place at Bank we offer a Retirement Fund service! You will be able to withdraw your money in 10 years!	Deploy
Challenge 4: Assume Ownership	WARNING! This is a restricted area! Only the Admin have access to it. Please, stay out of the way!	Deploy

Extensi3n: (MetaMask) - MetaMask Notification...

My Wallet

0x67B...2e3e

New address detected! Click here to add to your address book.

http://ethereum-web

0x67B...2e3e : CONTRACT INTERACTION

1

DETAILS DATA HEX

Estimated gas fee

0.00580994 0.00581 ETH

Max fee: 0.00580994 ETH

Total

1.00580994

1.00580994 ETH

Amount + gas fee

Max amount: 1.00580994 ETH

Reject

Confirm

This ether is now on the contract balance:

## Lottery!

Contract balance: 1 ether

The Bank is offering a price for a lottery game. Would you be able to guess the winner number? You only need to pay 1 eth each time you participate!

Condition to win: Get the balance of the contract to 0.

Enter number:

0



Guess!

Contract deployed at: [0xa9f4d51958918bd39d42fa7707c9a72cd02af855](#)

Check Solution

Challenge is NOT completed!



Now we have to manage a way to retrieve back the ether!

If you pay attention to the web page you will find the source code of all the challenges in /src folder:

← → ↻ ethereum-web/src/smart-contracts/

Por favor, ayúdanos a mejorar Firefox contestando esta breve encuesta [Descubre más](#) [Responde](#)

# Index of /src/smart-contracts

<a href="#">Name</a>	<a href="#">Last modified</a>	<a href="#">Size</a>	<a href="#">Description</a>
 <a href="#">Parent Directory</a>		-	
 <a href="#">AssumeOwnershipChallenge.sol</a>	2022-05-26 10:23	304	
 <a href="#">LotteryChallenge.sol</a>	2022-05-26 10:23	499	
 <a href="#">RetirementFundChallenge.sol</a>	2022-05-26 10:23	1.1K	
 <a href="#">TokenSaleChallenge.sol</a>	2022-05-26 10:23	719	

Having the source code is crucial to interact with the Smart Contract on Remix IDE and execute our exploit on it.

← → ↻ ethereum-web/src/smart-contracts/LotteryChallenge.sol

```
pragma solidity ^0.4.21;

contract LotteryChallenge {
    uint8 answer;

    constructor() public payable {
        require(msg.value == 1 ether);
        answer = uint8(keccak256(blockhash(block.number - 1), now));
    }

    function isComplete() public view returns (bool) {
        return address(this).balance == 0;
    }

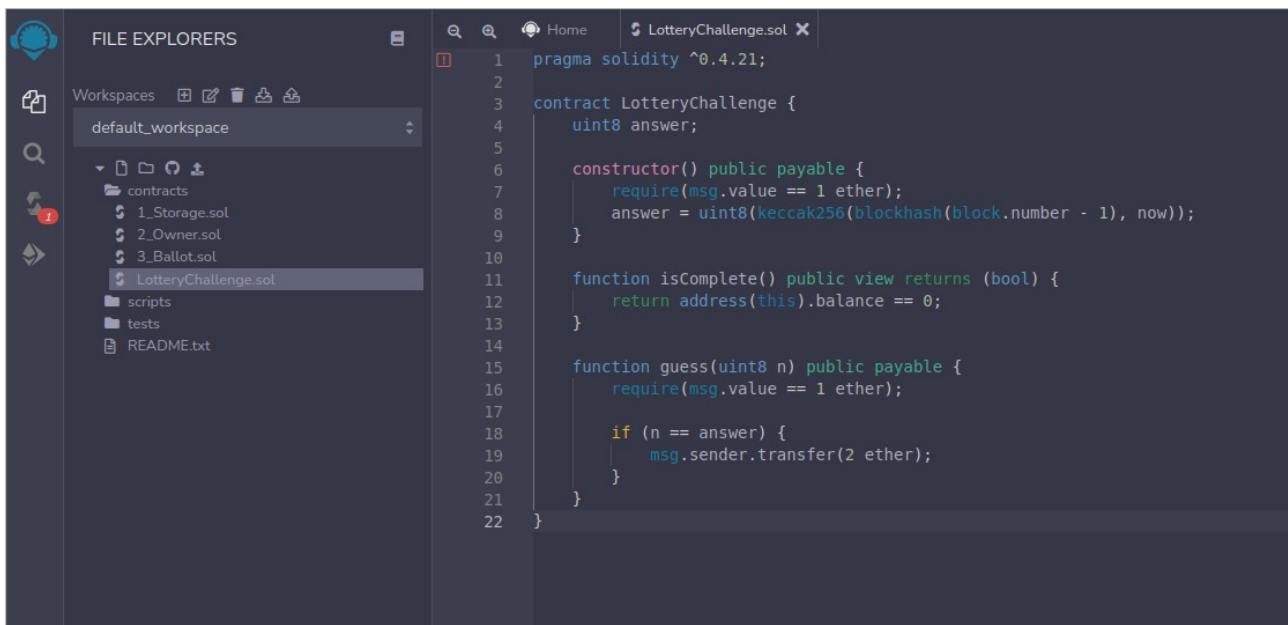
    function guess(uint8 n) public payable {
        require(msg.value == 1 ether);

        if (n == answer) {
            msg.sender.transfer(2 ether);
        }
    }
}
```

Let's open Remix IDE:

<https://remix.ethereum.org/>

Here you have a full development IDE online for Solidity. To interact with your contract first you need to compile the code, so you have to copy the code and create a file in the contracts folder on the IDE:

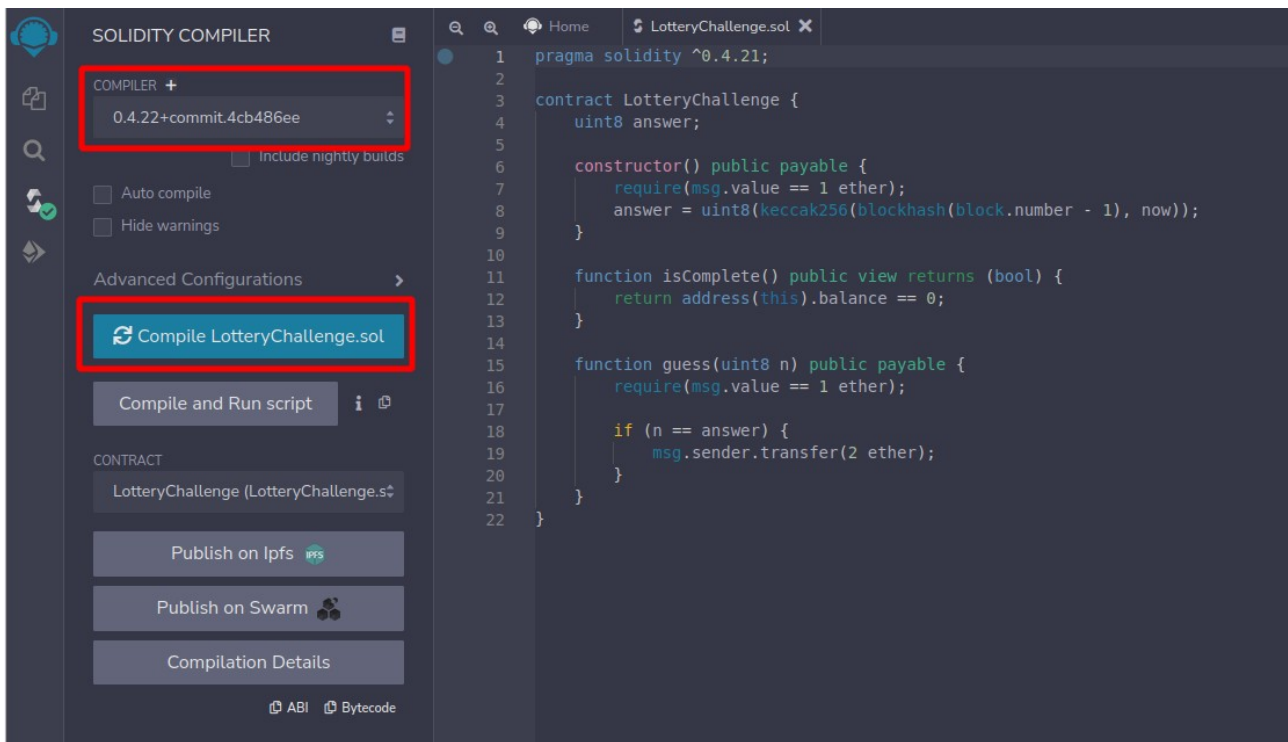


Next step is compiling the Smart Contract. If you pay attention, the code is made for compiler version 0.4.1.21 and above:

`pragma solidity ^0.4.21;`

So you have to navigate to compiler section, select the proper compiler version, and click on compile. (Note: compiler 0.4.21 may not work, use 0.4.22 instead).



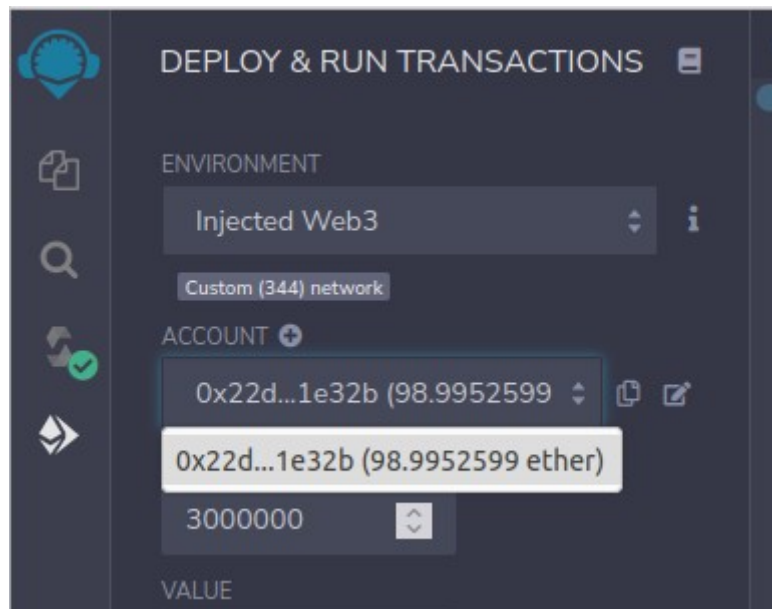


Ok, now we have the bytecode of the challenge compiled, but we have to tell Remix IDE where is it deployed in our CTF environment.

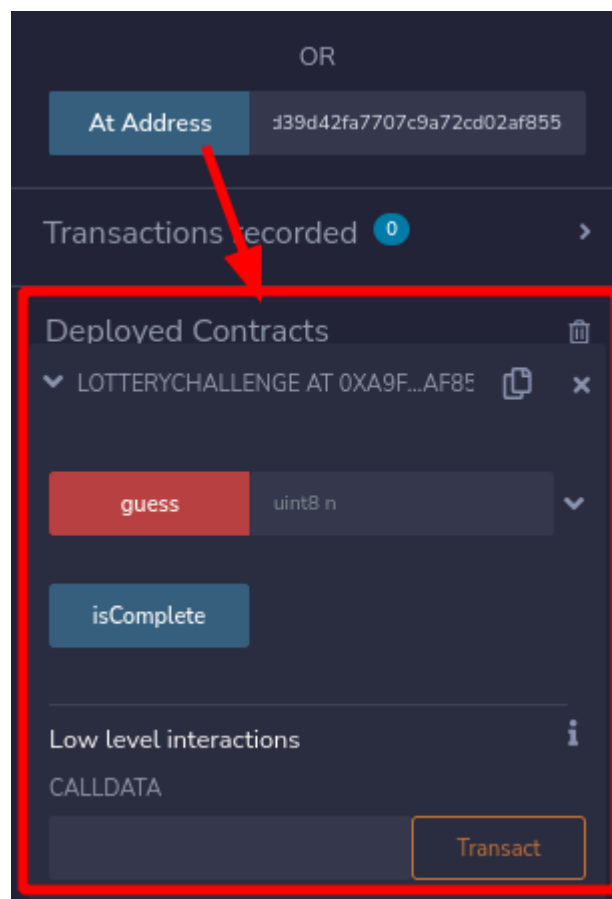
You should select the environment as “Injected Web3”. This tells Remix IDE that the interaction is performed through our Metamask wallet, out of the sandbox context provided by Remix IDE. Then, you will be prompted to connect your account to Remix IDE service:



If everything went ok, you should see your wallet account with your money:

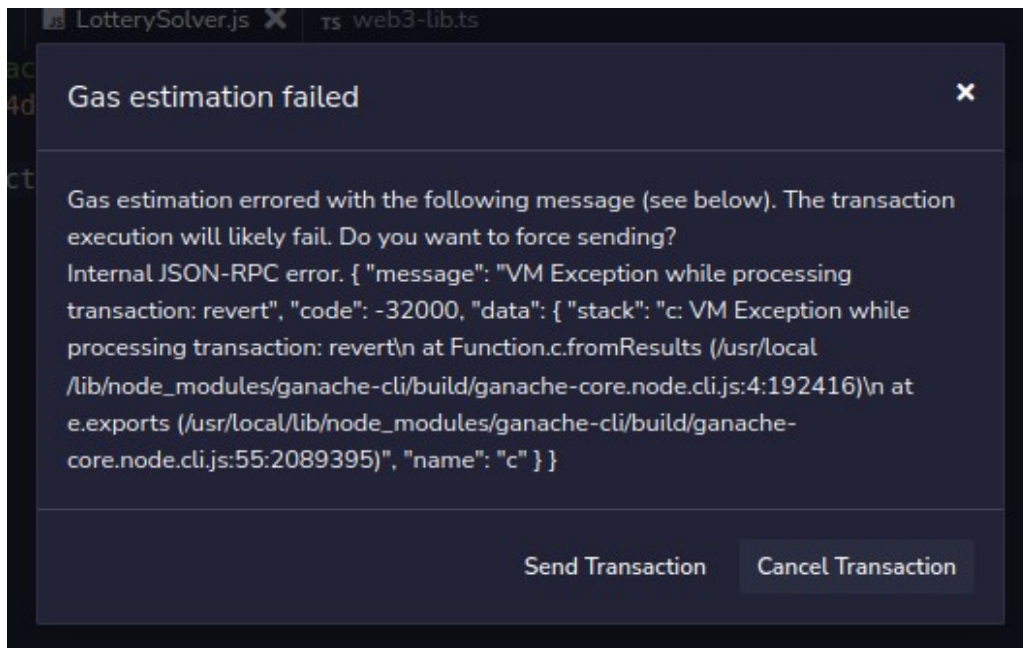


Now, we don't want to deploy a new challenge, but we want to use the one that already is deployed! So, you have to copy the address of the challenge and paste it on Remix IDE, and press the button "At Address":



Now you can start interacting with the contract from Remix IDE, and make a call to its public functions.

If you try to guess a number here it will give you an error. Why?



It's because the function is expecting that you pay 1 eth as requirement:

```
function guess(uint8 n) public payable {  
    require(msg.value == 1 ether);  
    if (n == answer) {  
        msg.sender.transfer(2 ether);  
    }  
}
```

You can tell Remix IDE that you want to pay  $10^{18}$  wei, or 1 ether, modifying the value:

The image shows the 'DEPLOY & RUN TRANSACTIONS' interface in Remix IDE. The 'ENVIRONMENT' is set to 'Injected Web3'. The 'ACCOUNT' is '0x22d...1e32b (98.9952599)'. The 'GAS LIMIT' is '3000000'. The 'VALUE' field is highlighted with a red box, showing '1' and 'Ether'. The 'CONTRACT' is 'LotteryChallenge - contracts/LotteryC'. There is a 'Deploy' button and a 'Publish to IPFS' checkbox. Below, there is an 'At Address' field with the address 'j39d42fa7707c9a72cd02af855'. The 'Transactions recorded' section shows '1' transaction. The 'Deployed Contracts' section shows 'LOTTERYCHALLENGE AT 0XA9F...AF85'. There is a 'guess' button and a '34' input field. Below that is an 'isComplete' button and the text '0: bool: false'.

You can now enter a number and click on the “guess” button, and interact with the contract!

In the challenge we can see now that the balance has been increased 1 more ether:

## Lottery!

Contract balance: 2 ether

The Bank is offering a price for a lottery game. Would you be able to guess the winner number? You only need to pay 1 eth each time you participate!

Condition to win: Get the balance of the contract to 0.

Enter number:

0

Guess!

Contract deployed at: [0xa9f4d51958918bd39d42fa7707c9a72cd02af855](#)

Check Solution

Challenge is NOT completed!

You can now take advantage of the many features that Remix IDE has! Try to change the fields and the values, write a PoC using Web3.js or create another Smart Contract that interacts with the challenge. Use your imagination to solve all the challenges. Good luck!