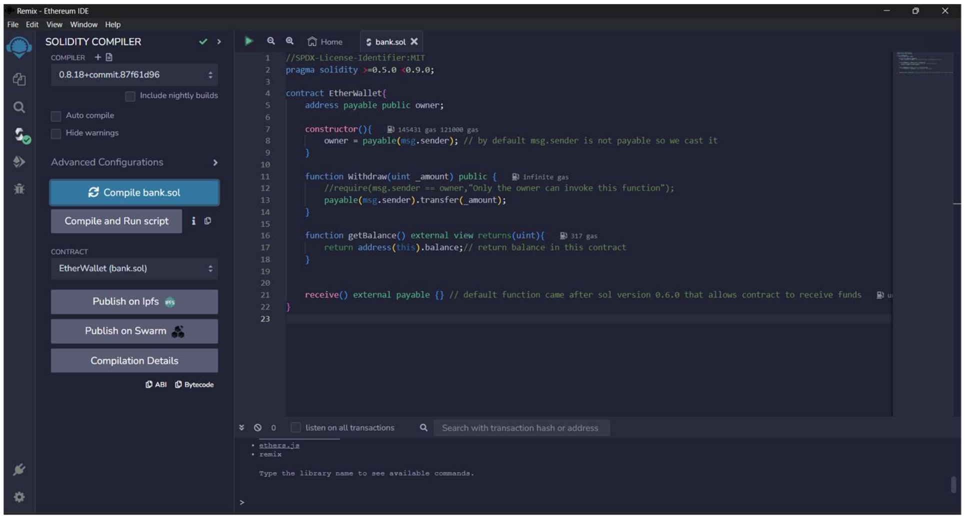
Code:

```
//SPDX-License-Identifier:MIT
pragma solidity >=0.5.0 <0.9.0;
contract EtherWallet {
address payable public owner;
constructor () {
      owner = payable(msg.sender); // by default msg.sender is not payable so we cast it
   }
function Withdraw (uint _amount) public {
      //require (msg.sender == owner,"Only the owner can invoke this function");
      payable(msg.sender).transfer(_amount);
   }
function getBalance() external view returns(uint){
      return address(this). balance;// return balance in this contract
   }
  receive () external payable {} // default function came after sol version 0.6.0 that allows contract to receive funds
}
```

Output:

Step 1: Compile



COMPILER: 0.8.18+commit.87f61d96

□ Auto compile

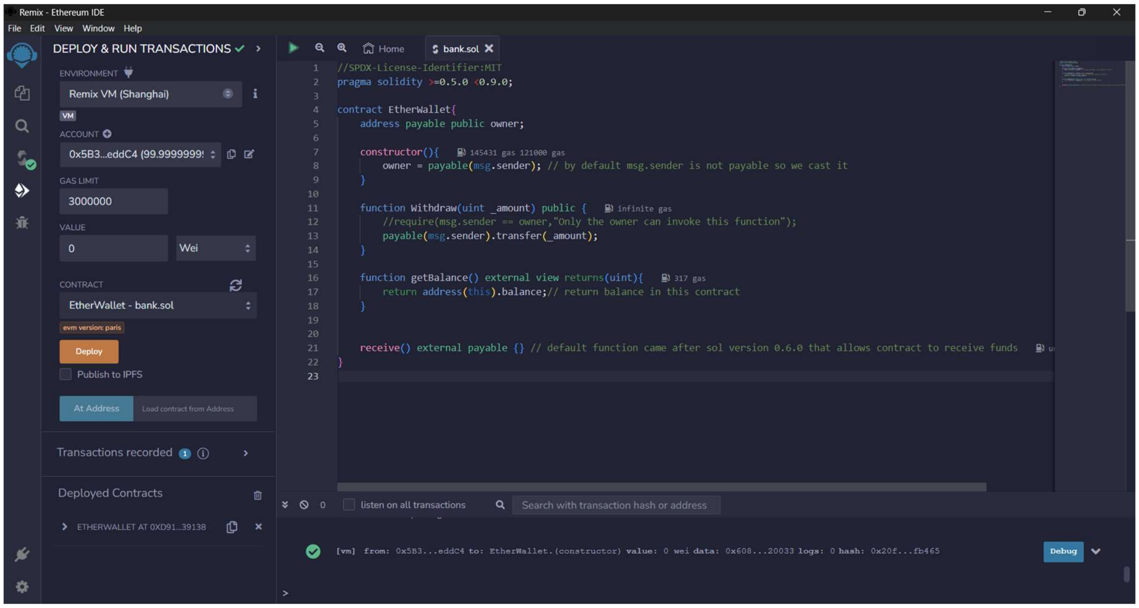
□ Hide warnings

CONTRACT: EtherWallet (bank.sol)

```
//SPDX-License-Identifier:MIT
pragma solidity >=0.5.0 <0.9.0;

contract EtherWallet{
    address payable public owner;

    constructor(){      145431 gas 121000 gas
        owner = payable(msg.sender); // by default msg.sender is not payable so we cast it
    }

    function Withdraw(uint _amount) public {      infinite gas
        //require(msg.sender == owner,"Only the owner can invoke this function");
        payable(msg.sender).transfer(_amount);
    }

    function getBalance() external view returns(uint){      317 gas
        return address(this).balance;// return balance in this contract
    }


    receive() external payable {} // default function came after sol version 0.6.0 that allows contract to receive funds
}
```

```
0
• ethers.js
• remix

  Type the library name to see available commands.

>
```

Step 2: Deploy



ENVIRONMENT: Remix VM (Shanghai)

VM

ACCOUNT: 0x5B3...eddC4 (99.9999999!

GAS LIMIT: 3000000

VALUE: 0 Wei

CONTRACT: EtherWallet - bank.sol

evm version: paris

Deploy

□ Publish to IPFS

At Address | Load contract from Address

Transactions recorded 1

Deployed Contracts

ETHERWALLET AT 0XD91...39138

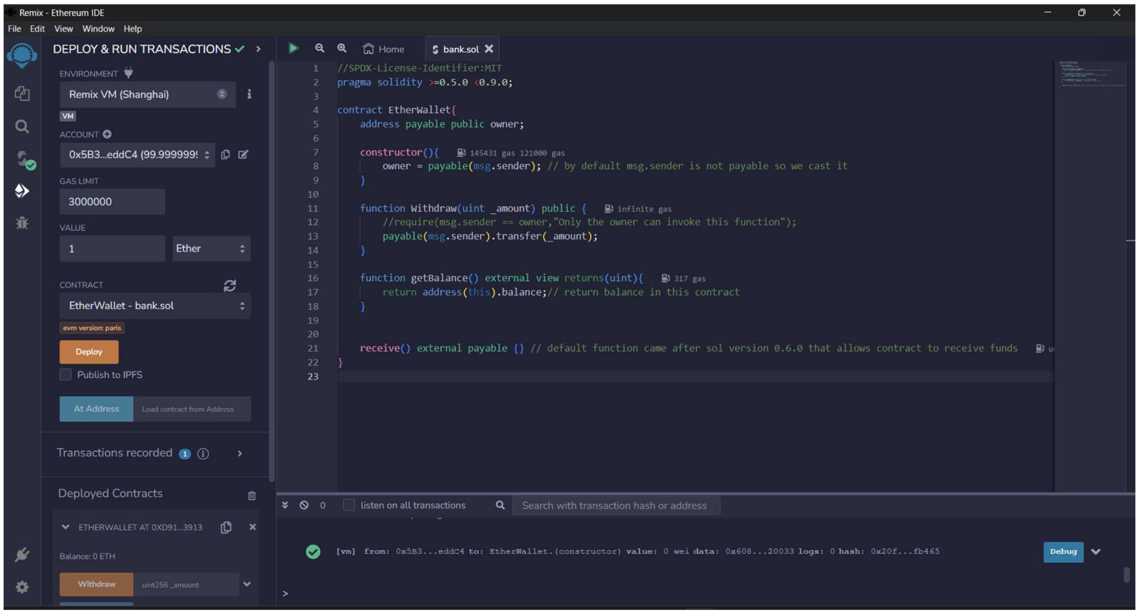
```
//SPDX-License-Identifier:MIT
pragma solidity >=0.5.0 <0.9.0;

contract EtherWallet{
    address payable public owner;

    constructor(){
        owner = payable(msg.sender); // by default msg.sender is not payable so we cast it
    }

    function Withdraw(uint _amount) public {
        //require(msg.sender == owner,"Only the owner can invoke this function");
        payable(msg.sender).transfer(_amount);
    }

    function getBalance() external view returns(uint){
        return address(this).balance;// return balance in this contract
    }


    receive() external payable {} // default function came after sol version 0.6.0 that allows contract to receive funds
}

```

□ listen on all transactions

Search with transaction hash or address

```
[vm] from: 0x5B3...eddC4 to: EtherWallet.(constructor) value: 0 wei data: 0x608...20033 logs: 0 hash: 0x20f...fb465
```

Step 3: Enter amount in Either



ENVIRONMENT: Remix VM (Shanghai)

VM

ACCOUNT: 0x5B3...eddC4 (99.999999!

GAS LIMIT: 3000000

VALUE: 1 Ether

CONTRACT: EtherWallet - bank.sol

evm version: paris

Deploy

□ Publish to IPFS

At Address | Load contract from Address

Transactions recorded 1

Deployed Contracts

ETHERWALLET AT 0XD91...3913

Balance: 0 ETH

Withdraw | uint256 _amount

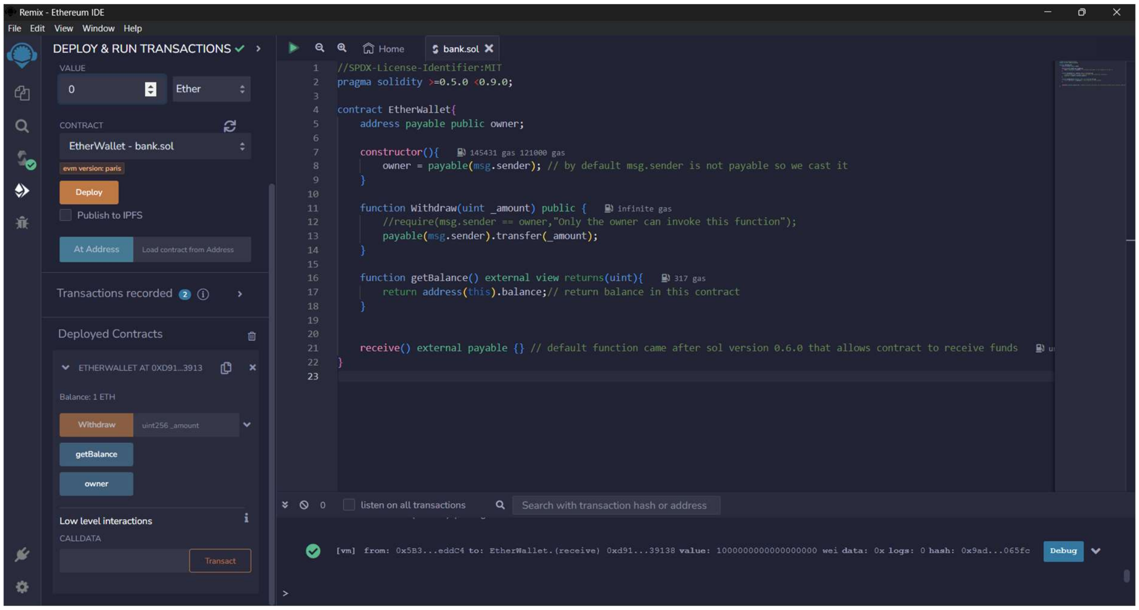
```
//SPDX-License-Identifier:MIT
pragma solidity >=0.5.0 <0.9.0;

contract EtherWallet{
    address payable public owner;

    constructor(){
        owner = payable(msg.sender); // by default msg.sender is not payable so we cast it
    }

    function Withdraw(uint _amount) public {
        //require(msg.sender == owner,"Only the owner can invoke this function");
        payable(msg.sender).transfer(_amount);
    }

    function getBalance() external view returns(uint){
        return address(this).balance;// return balance in this contract
    }


    receive() external payable {} // default function came after sol version 0.6.0 that allows contract to receive funds
}

```

□ listen on all transactions

Search with transaction hash or address

```
[vm] from: 0x5B3...eddC4 to: EtherWallet.(constructor) value: 0 wei data: 0x608...20033 logs: 0 hash: 0x20f...fb465
```

Step 4: Click on the Transact button on the bottom of contract.



VALUE: 0 Ether

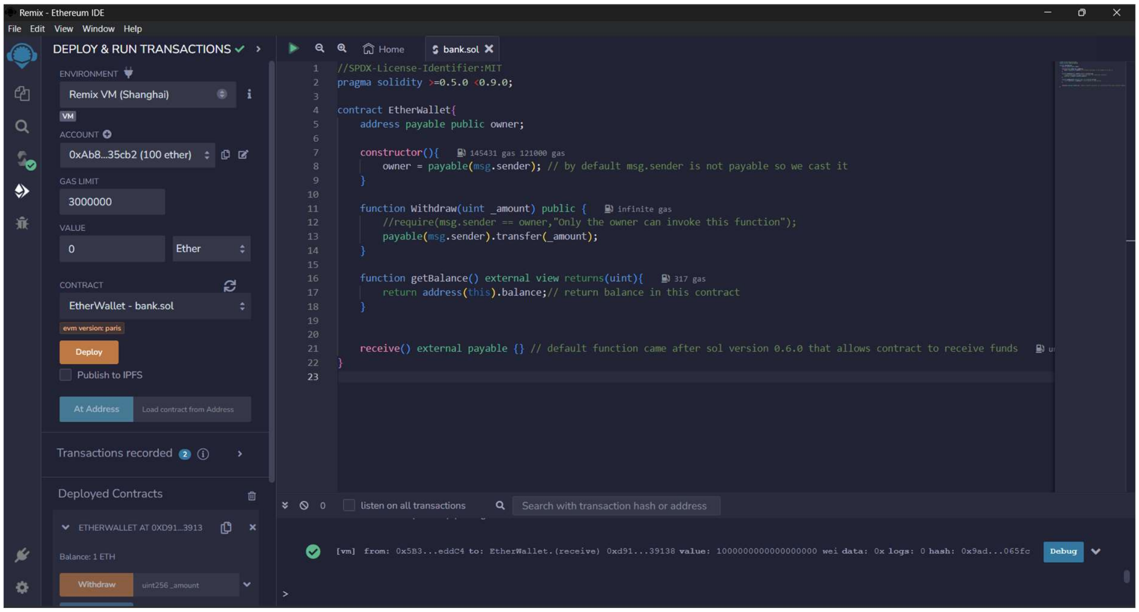
CONTRACT: EtherWallet - bank.sol

Balance: 1 ETH

```
//SPDX-License-Identifier:MIT
pragma solidity >=0.5.0 <0.9.0;

contract EtherWallet{
    address payable public owner;

    constructor(){    145431 gas 121000 gas
        owner = payable(msg.sender); // by default msg.sender is not payable so we cast it
    }

    function Withdraw(uint _amount) public {    infinite gas
        //require(msg.sender == owner,"Only the owner can invoke this function");
        payable(msg.sender).transfer(_amount);
    }

    function getBalance() external view returns(uint){    317 gas
        return address(this).balance;// return balance in this contract
    }


    receive() external payable {} // default function came after sol version 0.6.0 that allows contract to receive funds
}
```

```
[vm] from: 0x5B3...eddC4 to: EtherWallet.(receive) 0xd91...39138 value: 1000000000000000000 wei data: 0x logs: 0 hash: 0x9ad...065fc
>
```

Step 5: Select the other account


Remix - Ethereum IDE
File Edit View Window Help
DEPLOY & RUN TRANSACTIONS ✓
Home
bank.sol
VM
evm version: paris
Deploy
□ Publish to IPFS
At Address
Load contract from Address
Transactions recorded 2
Deployed Contracts
ETHERWALLET AT 0XD91...3913
Withdraw
uint256 _amount
listen on all transactions
Search with transaction hash or address
Debug

ENVIRONMENT: Remix VM (Shanghai)

ACCOUNT: 0xAb8...35cb2 (100 ether)

GAS LIMIT: 3000000

VALUE: 0 Ether

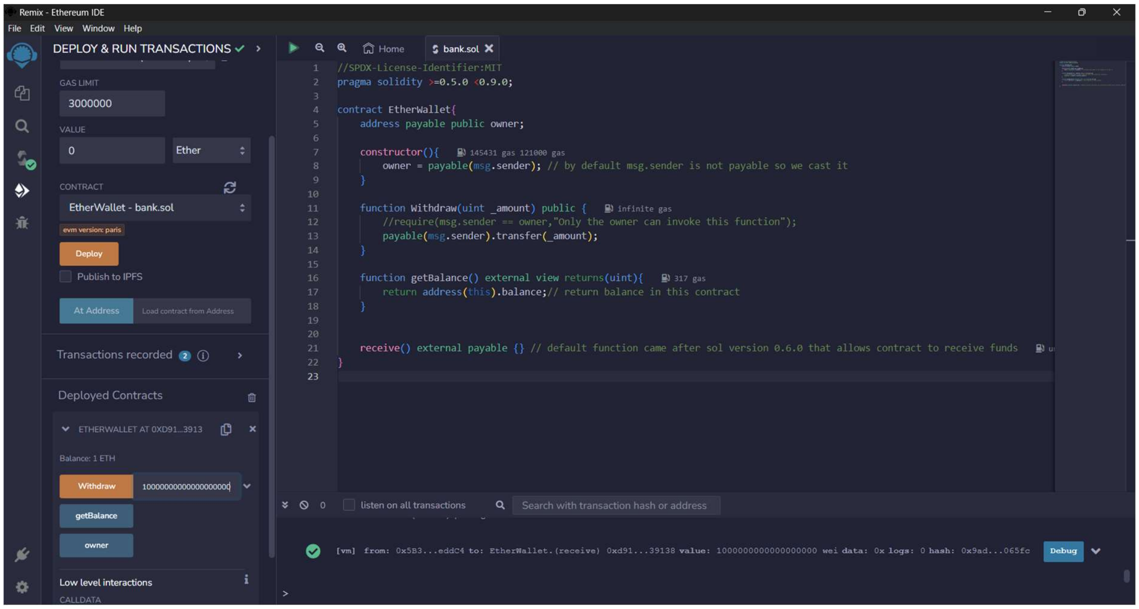
CONTRACT: EtherWallet - bank.sol

Balance: 1 ETH

```
//SPDX-License-Identifier:MIT
pragma solidity >=0.5.0 <0.9.0;

contract EtherWallet{
    address payable public owner;

    constructor(){    145431 gas 121000 gas
        owner = payable(msg.sender); // by default msg.sender is not payable so we cast it
    }

    function Withdraw(uint _amount) public {    infinite gas
        //require(msg.sender == owner,"Only the owner can invoke this function");
        payable(msg.sender).transfer(_amount);
    }

    function getBalance() external view returns(uint){    317 gas
        return address(this).balance;// return balance in this contract
    }


    receive() external payable {} // default function came after sol version 0.6.0 that allows contract to receive funds
}
```

```
[vm] from: 0x5B3...eddC4 to: EtherWallet.(receive) 0xd91...39138 value: 1000000000000000000 wei data: 0x logs: 0 hash: 0x9ad...065fc
>
```

Step 6: Click on the request amount in Wei i.e., 10^{18} zeros.



GAS LIMIT: 3000000

VALUE: 0 Ether

CONTRACT: EtherWallet - bank.sol

evm version: paris

Transactions recorded 2

Deployed Contracts

ETHERWALLET AT 0XD91...3913

Balance: 1 ETH

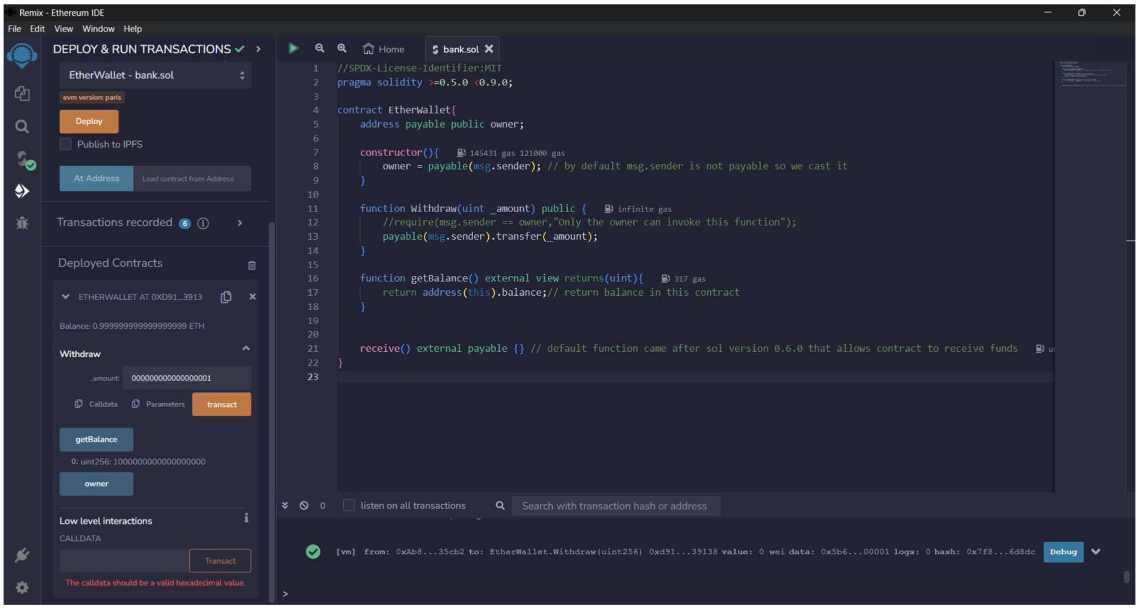
Withdraw: 1000000000000000000

```
//SPDX-License-Identifier:MIT
pragma solidity >=0.5.0 <0.9.0;

contract EtherWallet{
    address payable public owner;

    constructor(){    145431 gas 121000 gas
        owner = payable(msg.sender); // by default msg.sender is not payable so we cast it
    }

    function Withdraw(uint _amount) public {    infinite gas
        //require(msg.sender == owner,"Only the owner can invoke this function");
        payable(msg.sender).transfer(_amount);
    }

    function getBalance() external view returns(uint){    317 gas
        return address(this).balance;// return balance in this contract
    }


    receive() external payable {} // default function came after sol version 0.6.0 that allows contract to receive funds
}
```

0
listen on all transactions
Search with transaction hash or address

```
[vm] from: 0x5B3...eddC4 to: EtherWallet.(receive) 0xd91...39138 value: 1000000000000000000 wei data: 0x logs: 0 hash: 0x9ad...065fc
```

Step 7: Withdraw the amount of Rs. 0000000000000000001 and click on Transact.



EtherWallet - bank.sol

evm version: paris

Transactions recorded 6

Deployed Contracts

ETHERWALLET AT 0XD91...3913

Balance: 0.999999999999999999 ETH

Withdraw

_amount: 0000000000000000001

0: uint256: 1000000000000000000

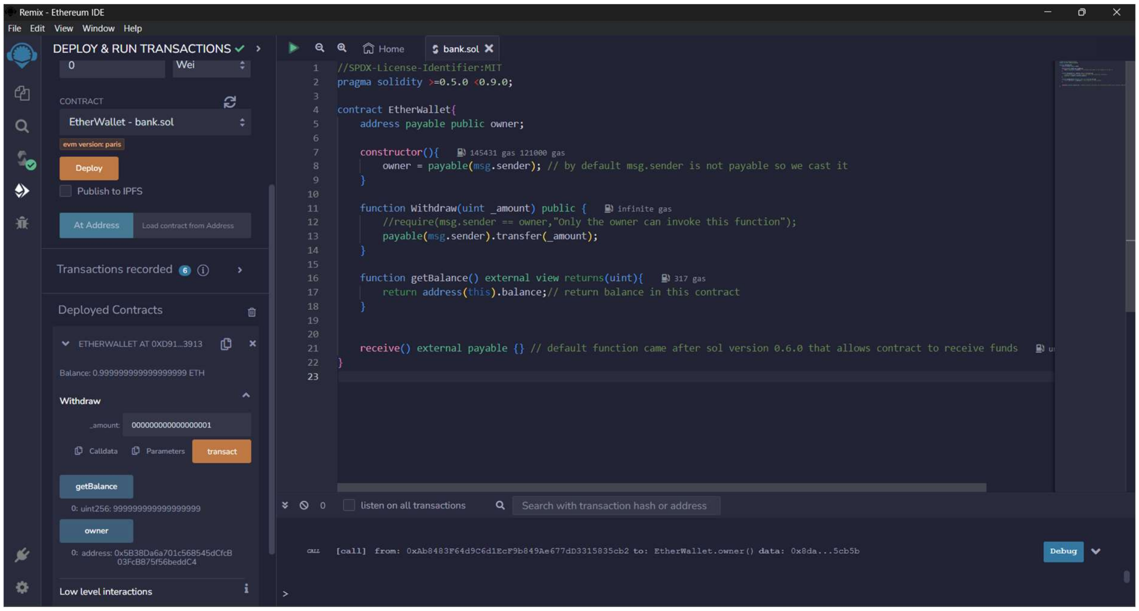
The calldata should be a valid hexadecimal value.

```
//SPDX-License-Identifier:MIT
pragma solidity >=0.5.0 <0.9.0;

contract EtherWallet{
    address payable public owner;

    constructor(){    145431 gas 121000 gas
        owner = payable(msg.sender); // by default msg.sender is not payable so we cast it
    }

    function Withdraw(uint _amount) public {    infinite gas
        //require(msg.sender == owner,"Only the owner can invoke this function");
        payable(msg.sender).transfer(_amount);
    }

    function getBalance() external view returns(uint){    317 gas
        return address(this).balance;// return balance in this contract
    }


    receive() external payable {} // default function came after sol version 0.6.0 that allows contract to receive funds
}
```

0
listen on all transactions
Search with transaction hash or address

```
[vm] from: 0xAb8...35cb2 to: EtherWallet.Withdraw(uint256) 0xd91...39138 value: 0 wei data: 0x5b6...00001 logs: 0 hash: 0x7f8...6d8dc
```

Step 8: Click on the getBalance to check the balance and click on the owner for owner name



0 Wei

CONTRACT: EtherWallet - bank.sol

evm version: paris

ETHERWALLET AT 0XD91...3913

Balance: 0.999999999999999999 ETH

Withdraw

_amount: 000000000000000001

0: uint256: 999999999999999999

0: address: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4

```
//SPDX-License-Identifier:MIT
pragma solidity >=0.5.0 <0.9.0;

contract EtherWallet{
    address payable public owner;

    constructor(){      145431 gas 121000 gas
        owner = payable(msg.sender); // by default msg.sender is not payable so we cast it
    }

    function Withdraw(uint _amount) public {      infinite gas
        //require(msg.sender == owner,"Only the owner can invoke this function");
        payable(msg.sender).transfer(_amount);
    }

    function getBalance() external view returns(uint){      317 gas
        return address(this).balance;// return balance in this contract
    }


    receive() external payable {} // default function came after sol version 0.6.0 that allows contract to receive funds
}
```

```
CALL [call] from: 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2 to: EtherWallet.owner() data: 0x8da...5cb5b
>
```